VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ZAKLÁDACÍ MANIPULÁTOR PRO 3D TISKÁRNU
THE CRANE MANIPULATOR FOR THE 3D PRINTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Khmil Denys
AUTHOR

VEDOUCÍ PRÁCE — Ing. Jakub Arm
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Denys Khmil ***ID:*** 200913

Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Zakládací manipulátor pro 3D tiskárnu

POKYNY PRO VYPRACOVÁNÍ:

Pro danou 3D tiskárnu navrhněte a realizujte (softwarová simulace) pomocí vhodných komponent manipulátor, který bude vybírat a zakládat tiskové podložky z a do regálu. Účelem je automatizovat manipulaci tiskových podložek 3D tiskáren.

1) Vytvořte model manipulátoru pomocí 3D CAD.
2) Vyberte vhodné komponenty pro řízení a elektrifikaci.
3) Navrhněte schéma elektrického zapojení.
4) Vytvořte pohyblivý virtuální model.
5) Vytvořte a nakonfigurujte softwarové vybavení manipulátoru a jeho ovládání.
6) Otestujte funkčnost na pohyblivém virtuálním modelu manipulátoru a vyhodnoťte vlastnosti.

DOPORUČENÁ LITERATURA:

Patrick Hood-Daniel. Build Your Own CNC Machine. Apress, 2009. ISBN: 978-1430224891.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Jakub Arm

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Tato bakalářská práce se zabývá návrhem stroje pro obsluhováni 3D tiskáren. Během práce byly vybraný vhodné mechanické a elektronické součástky, vytvořen 3D model manipulátoru v programu Siemens NX12, navrhnuta schéma elektrického rozvodu a desky plošných spojů pro Beaglebone Black a Arduino. Pro ovládaní manipulátoru byla vtvořena klientská aplikace pro počítač a Python skript pro Beaglebone (Machinekit). Výsledná konstrukce stroje byla otestovaná pomoci virtuálního pohyblivého modelu v programu Siemens MCD.

Klíčová slova

Manipulátor, 3D Tiskárna, CAD, Macinekit, Beaglebone Black, Siemens MCD.

Abstract

The aim of this work is to design the architecture of a manipulator that takes the finished product out of the 3D printer. This article deals with the whole concept of the machine, its design in the CAD program Siemens NX12, selection of parts, software development and virtual machine simulating. The manipulator is driven by Machinekit operating system installed on a singleboard computer Beaglebone Black. To control the machine, a client PC application was developed.

Keywords

Manipulator, 3D Printer, CAD, Machinekit, Beaglebone Black, Siemens MCD.

Bibliografická citace:

KHMIL, Denys. Zakládací manipulátor pro 3D tiskárnu. Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/127025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Jakub Arm.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Zakládací manipulátor pro 3D tiskárnu jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 30. května 2020 ……

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Jakubu Armovi, za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 30. května 2020 ……

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

1 ÚVOD

V dnešní době je 3D tisk často používán ve výrobě, při prototypování a v domácnosti. Nevýhodou klasických 3D tiskáren ve výrobě je, že při tisku větších počtů prvků nemohou začít nový tisk bez účasti člověka. Při dokončení tisku potřebují, aby obsluha vytištěný přípravek odstranila a tím samým umožnila spouštění dalšího tisku.

Analýzou dosavadní situace v této oblasti bylo zjištěno, že existuje několik řešení daného problému. Prvním z nich je tisk na pásový dopravník, který se po jeho skončení otočí a uvolní prostor pro další tisk. Druhým je využití 6-osého robotického manipulátoru pro odběr tisku. Každé z daných řešení má své nevýhody. U prvního řešení je nevýhodou nepevná podložka pro tisk, což může způsobit problémy při tisku. U druhého řešení je nevýhodou vysoká cena a omezený počet tiskáren.

Tato práce se zabývá návrhem stroje, který přistupuje k tomuto problému jiným způsobem. Během práce bude navržen manipulátor, který na základě signálu od tiskáren odebere podložku s vytištěnými díly a umístí novou pro provedení dalšího tisku.

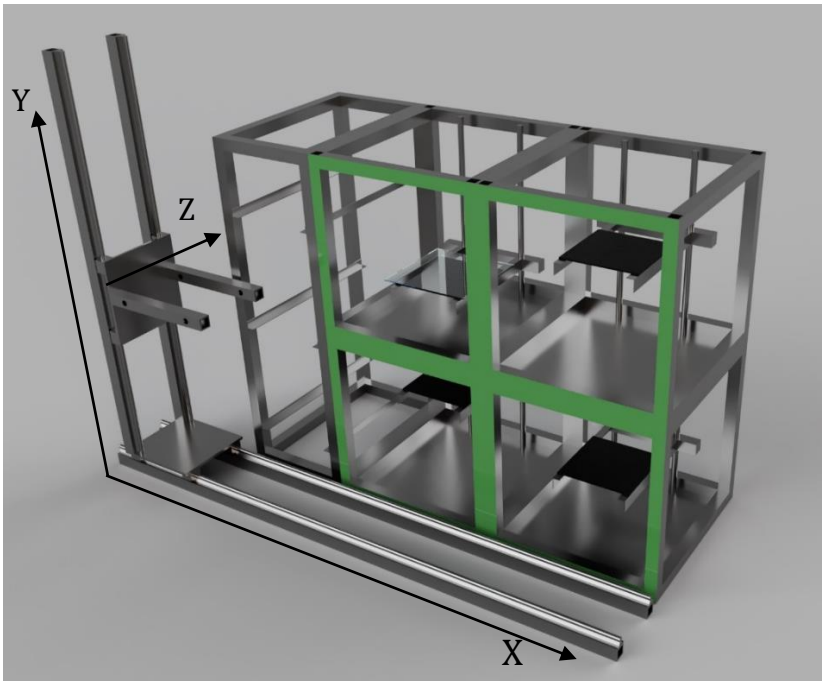
2 TEORETICKÝ ROZBOR

V dané kapitole bude vybrána hlavní koncepce manipulátoru a typy komponent, které budou použity pro další návrh a tvorbu prototypu, dále budou probrány jejich teoretické vlastnosti a princip funkčnosti.

2.1 Koncepce manipulátoru

Manipulátor bude navržen pro práce s tiskárnami o rozměru 500 × 500 × 560 mm, sklo o rozměru 300 × 300 mm umístěné na ohřívaném stole bude využito jako plocha pro tisk tiskárny. Tuto vlastnost je možné využít pro realizaci odběru vytištěného materiálu. Sklo bude odstraněno spolu s vytištěným materiálem a situováno do skladu, prázdné sklo bude poté umístěno do tiskárny a tím samým způsobem bude umožněno spouštění dalšího tisku.

Manipulátor bylo rozhodnuto realizovat jako trojosý CNC stroj, kde X je vodorovná osa, Y je svislá osa a osa Z slouží jako mechanismus odběru podložky pro tisk.



Obrázek 2.1 Koncepce manipulátoru

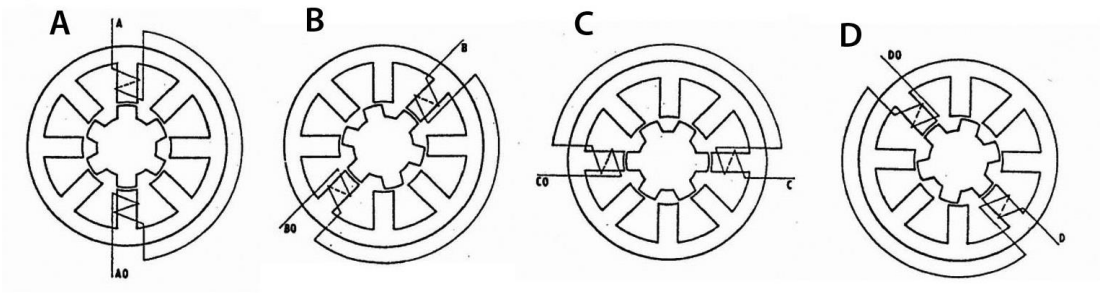
Na obrázku 2.1 je zobrazena základní koncepce manipulátoru a jsou označeny osy pohybu. Pro návrh byly změřeny rozměry 3D tiskárny a vytvořen orientační model, podle něhož byly určeny rozměry manipulátoru pro práci se čtyřmi tiskárnami. Získaná pracovní plocha je 1 × 1 m.

2.2 Princip činnosti krokových motorů

Krokový motor je synchronní motor (rotor se točí stejnou rychlostí jako točivé magnetické pole ve statoru). Točivé magnetické pole není vytvářeno střídavým proudem, ale postupným zapínáním jednotlivých cívek statoru.

Krokové motory je možné rozdělit do tří základních skupin:

- Krokové motory s pasivním rotorem – motory s vyjádřenými póly na statoru i rotoru, využívající výrazně rozdílné magnetické reluktance (vodivosti) v příčné i podélné ose.
- Krokové motory s aktivním rotorem – jejich rotor je tvořen permanentním magnetem. Podle uspořádání pólů magnetu odlišujeme dvě skupiny těchto motorů s radiálně polarizovaným nebo s axiálně polarizovaným permanentním magnetem.
- Krokové motory hybridní – slučují konstrukční principy obou předchozích typů.



Obrázek 2.2 Znázorněni pohybu rotoru krokového motoru [18]

Princip funkčnosti je zobrazen pro čtyřfázový motor s pasivním rotorem. Stator má 8 zubů a na každém je navinuta cívka. Dvojice cívek navinutých na protilehlých zubech jsou spojeny a tvoří vždy jednu fázi. Celkem máme 4 fáze – označeny A, B, C, D. Rotor má na svém povrchu 6 zubů. Otočení motoru bude realizováno tak, že bude postupně procházet proud různými fázemi, čímž vznikne magnetický tok mezi dvěma protilehlými cívkami. Tok bude procházet místem s nejmenším magnetickým odporem, což je v daném případě rotor, který se bude snažit zaujmout takovou polohu, kdy bude protékat maximální magnetický tok. Důležitý je rozdílný počet zubů rotoru a statoru, protože tehdy dochází k tomu, že se pouze dvojice zubů rotoru mohou překrývat se zuby statoru. Při každém přepínání cívky se bude muset rotor otočit do stavu s minimálním magnetickým odporem [18].

3 VYBER KOMPONENT

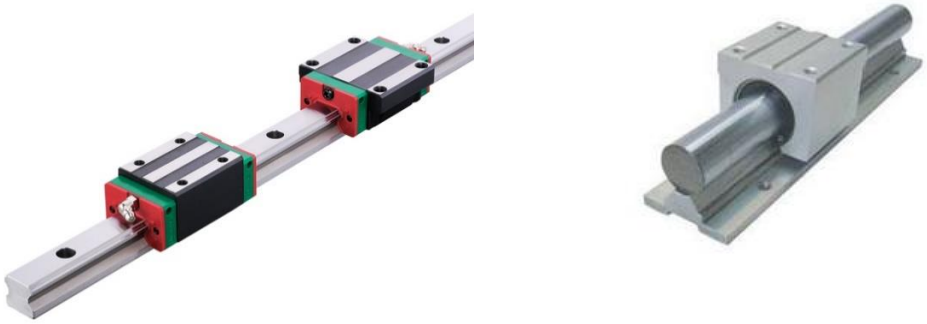
V dané kapitole bude proveden rozbor různých typů mechanických a elektrických komponent, budou probrány výhody a nevýhody jejich použití a rozhodnuto, které jsou nejvhodnější pro stavbu manipulátoru. Dále budou provedeny nutné výpočty a záměry.

3.1 Mechanika

3.1.1 Lineární vedeni

Po určení základní koncepce a rozměrů manipulátoru bylo důležité rozhodnout, jak bude realizován lineární pohyb vozíku osy X a osy Y. Hlavním požadavkem byla možnost co nejjednoduššího rozšíření pracovní oblasti manipulátoru a v případě potřeby vysoká přesnost a co nejmenší hmotnost.

Pro dané požadavky bylo možné vybrat ze dvou možností: lineární prizmatické vedení řady Hiwin HGR a lineární vedení typu podepřené tyče SBR.



a) b)

Obrázek 3.1 Lineární vedeni Hiwin HGR15 (a)[2] a SBR16 (b)[3]

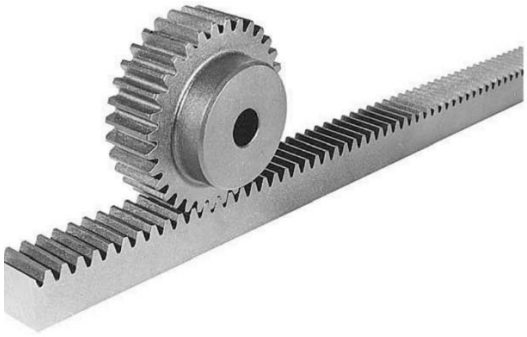
Na začátku návrhu bylo rozhodnuto využít lineární SBR16 z ekonomického hlediska, ale při dalším návrhu konstrukce bylo jasné, že jejich využití vyvolává nemalé těžkosti. Hlavním nedostatkem je montáž pomocí dvou řad šroubů, po jednom z každé strany tyče. Taková montáž nutí využívat hliníkové profily 30 × 60 mm a v místech montáže ozubeného hřebene 30 × 90 mm. Zmíněné profily zvětší celkovou hmotnost konstrukce a jsou dražší než 30 × 30 mm. Proto bylo SBR16 vyměněno za vedení Hiwin HGR. Pro určení rozměrů vedení byla vypočítána orientační hmotnost konstrukce (viz výpočet hmotnosti v kapitole 3.2.1) a z datasheetu vybrán vhodný rozměr vedení. Pro danou konstrukci je postačující

využití nejmenšího HGR15. Takovým způsobem se podařilo téměř dvakrát zmenšit hmotnost manipulátoru a vyhnout se využití velkých a drahých hliníkových profilů.

3.1.2 Lineární posun

Pro CNC stroje takového typu existuje několik možností realizace lineárního posunu. Nejvyužívanějšími jsou kuličkové šrouby a ozubené hřebeny.

V případě pohybu na velké vzdálenosti je nejvhodnější využití ozubeného hřebene a ozubeného kola. Použití takového způsobu dává možnost jednodušeji rozšířit pracovní oblast manipulátoru v případě zvětšení počtu tiskáren. Pro daný stroj byl vybrán ozubený hřeben o rozměru 15 × 15 mm a modulu 1. Takový hřeben zajistí dobrou přesnost pohybu.



Obrázek 3.2 Ozubeny hřeben a ozubené kolo [4]

Výběr ozubeného kola byl proveden podle těchto požadavků: modul 1 pro kompatibilnost s ozubeným hřebenem. Průměr vnitřního otvoru 5 mm pro kompatibilnost s hřídelem převodovky. Co nejmenší vnější průměr, aby minimalizoval ztráty krouticího momentu motoru na převodu. Z předložených možností byla vybrána varianta s počtem zubů 12 a vnějším průměrem 14 mm.

3.1.3 Řemenová převodovka

Pro osy X a Y jsou použity řemenové převodovky. Jejich využití je vynuceno tím, že je potřebné umístit motory tak, aby neomezovaly minimální výšku pohybu svislou osou. S využitím převodovky lze umístit motor v takové vzdálenosti od ozubeného hřebene, aby nevadil pohybu. Podrobnější popis umístění převodovky je uveden v kapitole 4. Návrh 3D modelu. Další výhodou využití převodovky je možnost nastavení převodového poměru využitím řemenic s různými poloměry a stejným způsobem lze v případě potřeby většího krouticího momentu změnit převodový poměr převodovky a neměnit motor.

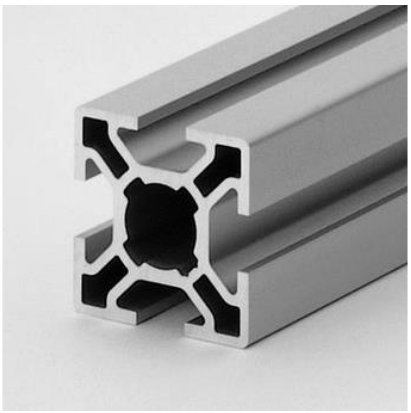
U manipulátoru jsou použity řemeny a řemenice HTD 5M se sirkou řemene 15 mm (viz Obrázek 3.3). Pro osu X má převodovka převodový poměr 1:1 a jsou tam

použity dvě řemenice s průměrem 22 mm a řemen o délce 295 mm. Pro osu Y bylo rozhodnuto realizovat převod 1 : 2, proto jsou použity řemenice s průměrem 22 mm a 44 mm, délka řemenu v takovém případě činí 350 mm.

Obrázek 3.3 Řemenová převodovka [5]

3.1.4 Hliníkové profily

Po určení komponent pro realizace lineárního vedení a lineárního posuvu je důležité vybrat, jakým způsobem bude nejjednodušší realizovat montáž rámu manipulátoru. Požadavky jsou: dobrá tuhost, jednoduché spojení profilů mezi sebou a možnost montáže dalších komponent k profilům.



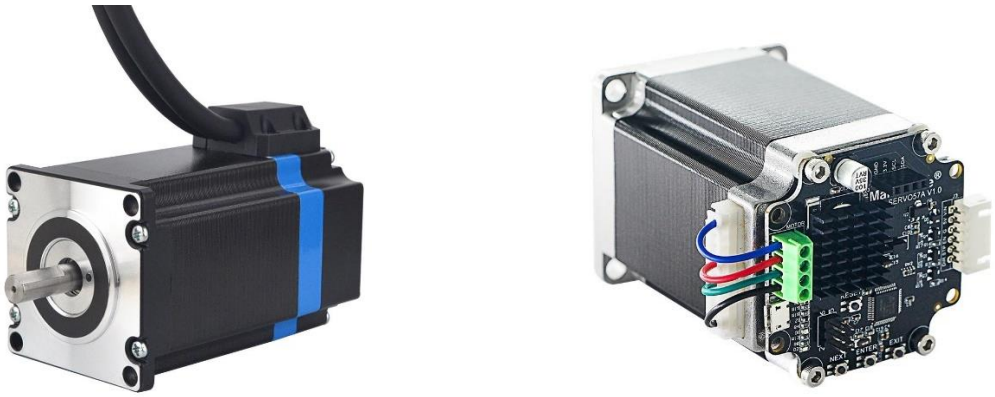
Obrázek 3.4 Hliníkový profil 30x30 [6]

Máme dvě možnosti: použít pro stavbu hliníkové jekly, nebo speciální stavebnicový profil (viz obrázek 3.4). Hliníkové jekly jsou levnější, ale vzniknou tak doplňkové náklady při montáži (vrtání, svařování atd.) a v případě potřeby už nepůjde konstrukce modifikovat. Stavebnicový profil má větší tuhost než jekly a pro jeho montáž existují speciální zámky, které umožní konstrukce kdykoliv modifikovat. Z tohoto hlediska je použití stavebnicového profilu vhodnější.

3.2 Motory

Pro pohyb v osách X a Y bylo rozhodnuto využít motory Nema23 se zpětnou vazbou. Při realizace zpětné vazby u takových motorů existuje několik způsobů. První variantou je průmyslové řešení, přičemž zpětná vazba bude realizována pomocí inkrementálního enkodéru. Další variantou je řešení pro hobby účely, přičemž by zpětná vazba byla realizována pomocí magnetu umístěného na hřídeli a bezkontaktních snímačů úhlu natočení. Takové motory existují ve verzích s různým krouticím momentem: 0.9N.m, 2N.m a 3N.m. Výpočet vhodného motoru je proveden v podkapitole 3.2.1 Výpočet vhodných motorů.

Pomocí zpětné vazby můžeme kontrolovat, aby během pohybu nedošlo ke ztrátě kroku motoru. V případě takové situace se driver pomocí hodnot z enkodéru detekuje a pokusí se akci zopakovat ještě několikrát. V momentu, kdy se mu to nepodaří, vyhodí chybu, která bude detekována řídicí elektronikou a zastaví pohyb celého manipulátoru. Takovým způsobem je možné zachránit stroj v situaci, kdy bude blokován pohyb jednoho z motorů a zbylé motory budou nadále pracovat.



a) b)

Obrázek 3.5: Nema23 s enkodérem (a)[7] a Nema23 se zpětnou vazbou pomoci magnetu (b)[8]

Pro stavbu prototypu budou z cenového hlediska použity motory se zpětnou vazbou pomoci magnetu, ale navržené schéma zapojení uvažuje o použití průmyslových motorů.

Pro pohyby v ose Z bylo rozhodnuto využití motoru Nema17, takové motory mají malý rozměr a hmotnost, ale nemají dostatečný krouticí moment pro realizaci pohybu dopravníku a vozíku osy Z. Alternativou může být využití DC motoru s převodovkou. Při použití daného manipulátoru má řešení dvě nevýhody: za prvé nebude umožněno řízení motoru na dálku, za druhé bude mít DC motor s převodovkou větší hmotnost než krokový motor. Z ekonomického hlediska je využití krokových motorů vhodnější.

3.2.1 Vypočet vhodných motorů

Pro výpočet správných motorů manipulátoru je nutné spočítat několik základních vlastností pohonu. Hlavní jsou potřebný krouticí moment motoru a žádoucí zrychlení, se kterým se bude motor pohybovat.

Nejprve je nutné určit hmotnosti všech komponent pro osu X a Y. S využitím datasheetu použitých komponent byly určeny jejich hmotnosti a vyneseny do tabulky č. 1 a tabulky č. 2.

Hodnoty uvedené v tabulkách jsou orientační, protože jsou vypočtené z hodnot uvedených v datasheetech. Na úkor toho bude v dalších výpočtech pro krouticí moment motoru přidána rezerva 20 %, která vykompenzuje rozdíl mezi reálnou a teoretickou hmotností.

Tabulka 3.1 Hmotnosti komponent pro vypočet motoru osy Y (vozík osy Z)

Komponent	Počet	Hmotnost
Motor Nema23	1	1.35 kg
Motor Nema17	2	0.7 kg
Vedeni Hiwin HGR15	750 mm	1.1 kg
Vozík HGR15	8	1.44 kg
Profil 30 x 30	3.8 m	3.5 kg
Hliníkové plechy	0.023 m^2	0.25 kg
Týc 8 mm	400 mm	0.16 kg
Šroub 8 mm	400 mm	0.12 kg
Max hmotnost tisku	1	0.5 kg

Tabulka 3.2 Hmotnosti komponent pro vypočet motoru osy X (vozík osy Y)

Komponent	Počet	Hmotnost
Motor Nema23	1	1.35 kg
Vedeni Hiwin HGR15	2 m	2.8 kg
Vozík HGR15	4	0.72 kg
Profil 30 x 30	3.7 m	3.33 kg
Profil 30 x 60	1 m	1.8 kg
Ozubeny hřeben	1 m	1.5 kg
Protizávaží	1	9 kg
Vozík osy Z	1	9.12 kg

Pro začátek musíme vypočítat celkovou hmotnost soustavy, dále pohyb, který bude zajištěn pomocí motoru. Toto lze vypočítat jako součet hmotností z tabulek 3.1 a 3.2 pro osu Y:

$$m_y = 9.12 \text{ [kg]} \quad (1)$$

Pro osu X:

$$m_x = 29.62 \text{ [kg]} \tag{2}$$

Dalším krokem je určení zrychlení, se kterým se bude sestava pohybovat. V daném případě rozhodneme, že maximální rychlost pohybu bude omezena na 0.4 m/s. Dále chceme, aby za 2 s dosahoval stroj maximální rychlosti. Dle těchto informací lze vypočítat zrychlení:

$$a = \frac{v - v_0}{t} = \frac{0.4 - 0}{2} = 0.2 \ [m/s^2] \tag{3}$$

Pro pohyby v ose Y bude použito protizávaží tak, aby vykompenzovalo hmotnost vozíku. Takové uspořádání umožní využití motoru s menším krouticím momentem a zjednoduší brzdění.

S využitím protizávaží bude hmotnost prázdného vozíku vykompenzována, proto motor osy Y musí mít dostatečný moment pro udržení hmotnosti vytištěného dílu a na zajištění pohybu s požadovaným zrychlením.

Maximální hmotnost vytištěného dílu je 0.5 kg, proto síla nutná pro udržení jeho hmotnosti je:

$$F_{m(y)} = m_y * g = 0.5 * 9.81 = 4.91 \ [N] \tag{4}$$

Sila, nutná pro zajištěni zrychleni 0.2 m/s^2:

$$F_{a(y)} = (m_{voz} + m_{zavaz}) * a = (9.12 + 8.62) * 0.2 = 3.55 \ [N] \tag{5}$$

Celková sila s rezervou 20%:

$$F_y = 1.2 * \left(F_{m(y)} + F_{a(y)}\right) = 1.2 * (4.91 + 3.55) = 10.15 \ [N] \tag{6}$$

Stejné výpočty musíme provést pro osu X, jediná odlišnost je v tom, že osa se nachází ve vodorovné poloze, a proto motor nemusí udržovat hmotnost sestavy, ale uplatni se tady sila třeni. Proto musíme tady vypočítat silu nutnou pro zajištěni zrychleni 0.2 m/s^2 a silu třeni.

$$F_{a(x)} = m_x * a = 29.62 * 0.2 = 5.92 \ [N] \tag{7}$$

Činitel třeni pro lineární vedeni Hiwin HGR15 byl stanoven z datasheetu a rovna se 0.004 [12]. Z toho můžeme vypočítat třecí sílu:

$$F_{tr(x)} = \mu * N = \mu * m_x * g = 0.004 * 29.62 * 9.81 = 1.16 \ [N] \tag{8}$$

$$F_x = 1.2 * \left(F_{a(x)} + F_{tr(x)}\right) = 1.2 * (5.92 + 1.16) = 8.5 \ [N] \tag{9}$$

Nyní, kdy jsou vypočtené všechny potřebné síly, můžeme spočítat moment, který budeme potřebovat při využití ozubeného kola o průměru 12 mm.

$$M_x = F_x * \frac{d}{2} * \sin(\alpha) = 8.5 * \frac{0.012}{2} * \sin(90) = 0.051 \ [N.m] \tag{10}$$

$$M_y = F_y * \frac{d}{2} * \sin(\alpha) = 10.15 * \frac{0.012}{2} * \sin(90) = 0.061 \ [N.m] \tag{11}$$

Je nutné uvažovat to, že pro osu Y je použita řemenová převodovka s převodovým poměrem 1:2, proto krouticí moment potřebný na hřídeli motoru bude dvakrát menší.

$$M_{y\_prev} = \frac{M_y}{2} = \frac{0.061}{2} = 0.0305\ [N.m] \quad (12)$$

Nejmenší motory Nema23 s enkodérem mají krouticí moment 0.9N.m, což je postačující k zajištění pohybu v osách X a Y. V tuto chvíli je zajištěna bezpečnost při ztrátě kroku a rezerva při případném rozšíření pracovní plochy.

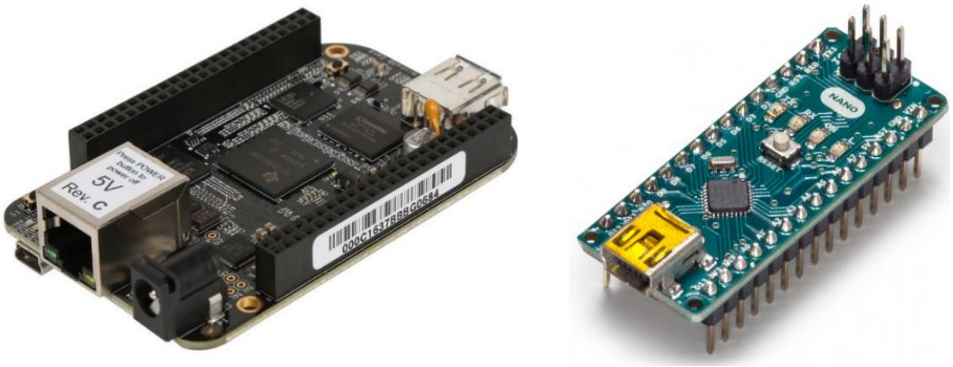
3.3 Elektronika

V dané kapitole budou vybrány komponenty k realizaci řízení stroje. Elektroniku můžeme rozdělit na hlavní a pomocný řídicí komponent, drivery pro motory, snímače a zdroje napájení.

3.3.1 Řídicí elektronika

Pod řídicí elektronikou se zde rozumí přístroj, který bude na základě příkazu uživatele a signálů ze snímačů řídit pohyb stroje a zapínat systém odběru vytištěného dílu z tiskárny.

Řídicí elektronika byla rozdělena na hlavní řídicí komponent a pomocný řídicí komponent. Hlavní řídicí komponent má za úkol řízení motoru osy X a Y, komunikaci s pomocným řídicím komponentem a také musí zajistit možnost realizace uživatelského rozhraní při dalším rozvoji projektu. Protože bylo rozhodnuto, že využití driveru pro motory pracující s protokolem EtherCAT a komunikace mezi hlavním a pomocným řídicím přístrojem bude realizována pomocí protokolu RS-485, musí mít podporu pro práci s oběma protokoly.



a) b)

Obrázek 3.6 Beaglebone Black(a) [9] a Arduino Nano(b) [15]

Pro dané požadavky můžeme použít stolní PC. Jako průmyslové řešení může být použit PLC, anebo jednodušší a levnější variantou může být využití mikropočítačů typu Beaglebone nebo Raspberry Pi. Bylo rozhodnuto, že z hlediska cen a jednoduchosti je aplikování mikropočítačů nejvhodnějším řešením. Zbývá tedy jenom vybrat mezi Rasbperry Pi a Beaglebone Black. Při porovnání bylo zjištěno,

že Beaglebone je lépe přizpůsoben pro využití ve strojích takového typu. Oproti Raspberry má větší počet vstupů a výstupů, podporuje analogové vstupy a PWM výstupy. Pro Beaglebone existují moduly pro realizaci CNC stroje. Z tohoto hlediska bylo rozhodnuto využít Beaglebone Black jako hlavní řídicí komponentu [15].

Pomocný řídicí přístroj bude použit pro řízení osy Z a mechanismus odběru tiskové podložky z tiskárny. Pomocí něj bude realizováno řízení dvou motorů Nema17 metodou step/dir na základě hodnot ze snímačů. Pro dané požadavky stačí využít Arduino, která má dostatečný počet vstupů a výstupů. Pomocí ní také lze pohodlně řídit motory, proto jako pomocný řídicí přístroj byla zvolena Arduino Nano.

3.3.2 Drivery pro motory

Drivery pro motory Nema23 byly vybrány z hlediska těchto kritérií: musí mít dostatečný výstupný proud, což jsou 2 A pro vybrané motory. Dále nesmí chybět podpora zpětné vazby pomocí enkodéru [16].

Drivery vyhovující takovým požadavkům se liší způsobem přivedení řídicího signálu, jsou tady dvě možnosti:

1. Řízeni pomoci step/dir signálu. U takových driveru potřebujeme pro pohyb přivést 4 signály:
    - Enable, který aktivuje napájení motoru.
    - Dir, který určuje směr otáčení motoru.
    - Step, každý pulz daného signálu bude odpovídat jednomu kroku motoru.
    - Zem, společná se zdrojem signálu.
2. Řízení pomocí průmyslových protokolů typu EtherCAT, Powerlink atd. Takové protokoly používají pro komunikaci modifikovaný protokol TCP/IP, přidávají k němu funkce diagnostiky, synchronizace a nové algoritmy.

Řízení pomocí step/dir signálu má několik nevýhod. Hlavní je nutnost mít vysokofrekvenční zdroj pulzu pro generace signálu step. Další nevýhodou je horší spolehlivost při větších vzdálenostech od zdroje signálu. Výhodou je nižší cena.

Pro průmyslové realizace stroje je vhodné použití driverů s podporou průmyslových protokolů. Pro stavbu prototypu, jak už bylo řečeno, budou použity motory se zpětnou vazbou pomocí magnetu a snímačů úhlu natočení. Drivery jsou u takových motorů kompletní a jsou umístěny na zadní straně motoru (viz obrázek 3.5b).

a) b)

Obrázek 3.7 Driver DM506-EC (a) [10] a TB6600 (b) [11]

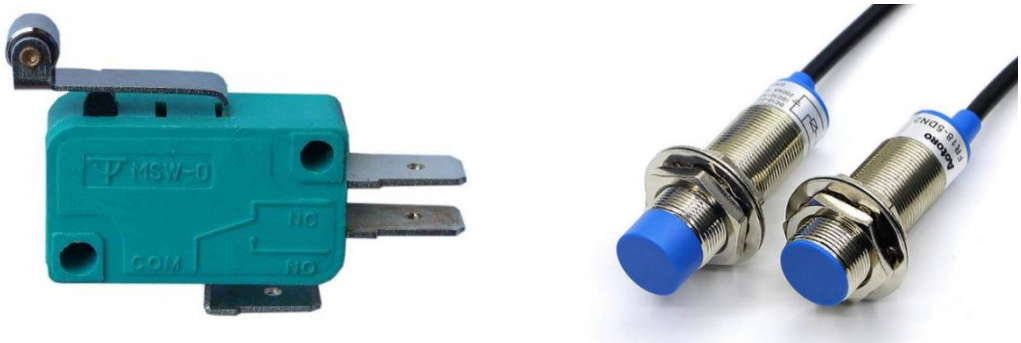
Při výběru driveru s podporou průmyslových protokolů bylo nalezeno existující řešení s podporou protokolů Powerlink, Modbus, EtherCAT a Profinet. Většina možností má vyšší cenu, nebo zbytečný maximální výkon. Ze všech možností byly vybrané drivery DM506-EC využívající protokol EtherCAT. Tyto drivery mají maximální proud 5.6 A, podporují zpětnou vazbu pomocí enkodéru a jsou levné.

Pro motory Nema17 byly vybrané drivery TB6600, protože jsou z hlediska ceny a spolehlivosti nejvhodnější.

3.3.3 Snímače

Pro návrh konstrukce daného stroje je potřeba několik typů snímačů, které lze rozdělit podle jejich funkce:

- Limitní snímače pro detekce nulové polohy.
- Snímače pro detekce podložky na dopravníku osy Z.
- Bezpečnostní snímače, které zajistí vypnutí napájení stroje v případě, že by došlo k překročení povolené délky projezdu.



a) b)

Obrázek 3.8 Koncový spínač (a)[19] a indukční snímač (b)[20]

Limitní snímače budou používané při každém spuštění stroje, proto musí být spolehlivé a musí mít dobrou trvanlivost. Nejjednodušším řešením by bylo použití koncových spínačů, ale jejich nevýhodou je, že když se často používají, snadno se opotřebují. Proto bylo rozhodnuto využití bezkontaktních indukčních snímačů, které budou detekovat přítomnost hliníkového vozíku na nulové pozici. Koncové spínače s rozpínacím kontaktem lze použít jako bezpečnostní snímače. V případě, že by pohyb nebyl zastaven pomocí indukčního snímače, vozík by narazil na koncový spínač a bezpečnostní obvod by se rozpojil. Tím samým způsobem by se vypnulo napájení.

Podložka pro tisk může být vyráběna z různých materiálů (sklo, hliník atd.), proto musíme vybrat takový snímač, který bude schopen detekovat potřebné materiály. Pro takové požadavky je vhodné využít kapacitní bezkontaktní snímač, který umožní detekovat podložku na dopravníku a nebude vadit jejímu pohybu.

3.3.4 Napájení

Napájení bylo rozděleno do dvou části:

- Napájení motorů Nema23 a driveru pro ně.
- Napájení řídicí elektroniky, snímačů a motorů Nema17.

Pro napájení motorů Nema23 existují dvě možnosti: napájení pomocí spínaného zdroje, nebo použití transformátorového zdroje.

Jako zdroj bylo rozhodnuto využít toroidní transformátor a modul usměrňovače. Použití transformátoru má v daném případě několik výhod: zdroje jsou spolehlivější a oproti spínaným zdrojům nemají na výstupu vysokofrekvenční šum, který může sloužit jako rušení pro driver. Tento transformátor je také vhodnější v případech, kdy může dojít k rychlému nárůstu proudu, což v daném případě nastane při začátku pohybu manipulátoru. Nevýhodou transformátoru je jeho větší rozměr a hmotnost, ale pro stacionární stroj to vadit nebude.

Obrázek 3.9 Toroidní transformátor s modulem usměrňovače [12]

Dané transformátory existují pro výstupní napětí 24 V, 36 V, 45 V, 55 V a 60 V. Zvolené motory mají největší krouticí moment při napájení napětím 48 V, proto byl vybrán transformátor s nejbližší hodnotou výstupního napětí, což je 45 V. Dalším kritériem je výběr maximálního proudu. Minimální proud, který musí transformátor zajistit, se bude skládat z největšího možného proudu, který vykoná motor (5 A), plus proud spotřeby driveru. Ztráty výkonu na driveru nejsou uvedeny v datasheetu, proto bude uvedena rezerva 15 % od maximálního proudu motoru, která vykompenzuje tyto ztráty.

$$I = 1.15 * I_{mot} = 5.75\ [A] \quad (13)$$

Maximální proud daných transformátorů je 7 A, což vyhovuje požadavkům.

Obrázek 3.10 Průmyslový spínaný napájecí zdroj 24V [13]

Zvolené elektronické komponenty mají rozlišné napájecí napětí, proto bylo nutné pro napájení elektroniky zvolit takový zdroj, který by vyhovoval většině elektronických součástek. Pro dané účely byl zvolen průmyslový spínaný zdroj s výstupním napětím 24 V. Z hlediska toho, že Beaglebone Black a Arduino Nano potřebují pro napájení nižší napětí, bylo rozhodnuto využít dvou stepdown modulů, které sníží napětí z 24 V na 5 V nutných pro napájení. Zbývá tedy jen rozhodnout o hodnotě maximálního proudu.

Motory Nema17 mají maximální proud 1.5 A, mikropočítač Beaglebone Black má spotřebu 10 W při napájení 5 V. Po převodu pomocí stepdown modulu by potřeboval dosáhnout maximální hodnoty 0.5 A. Arduino a snímače mají nízkou spotřebu, proto zvolíme reservu 1 A [6] [7].

Celkový proud je:

$$I = 2 * 1.5 + 0.5 + 1 = 4.5\ [A] \quad (14)$$

Z toho hlediska byl zvolen spínaný zdroj 24V 5A.

4 NÁVRH 3D MODELU

V dané kapitole bude proveden návrh 3D modelu manipulátoru s využitím komponent, které byly vybrány v kapitole č. 3. Návrh mechanické konstrukce byl proveden pomocí CAD programu Siemens NX 12. Během práce byly vytvořeny tři různé varianty manipulátoru, v dalších podkapitolách je uveden detailnější popis každé z nich.

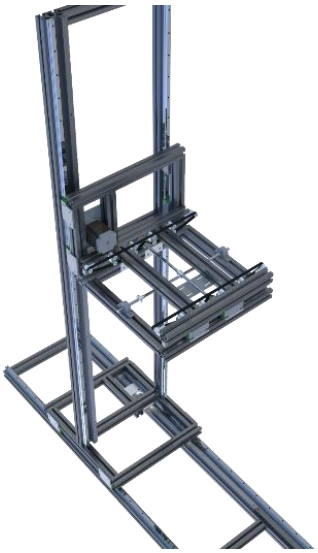
4.1 První varianta

První varianta manipulátoru je vidět na obrázku 4.1. Daná verze je přizpůsobena pro umístění na podlaze. Osy X a Y jsou zde realizovány jako dva profily, přičemž jeden z nich je 30 × 30 mm a druhý 30 × 60 mm. Do nich jsou umístěna lineární vedení a ozubený hřeben. Vedení osy X je umístěno na podlaze a na něm se pohybuje vozík osy Y s pevně umístěným vedením této osy.

Výhodou takového uspořádání je možnost použití manipulátoru jako samostatného stroje a možnost obsluhováni tiskáren na obou stranách manipulátoru.

Nevýhodou takového uspořádání oproti dalším variantám je větší počet profilů, což zapříčiní zvýšení ceny. Kvůli tomu, že není manipulátor upevněn v horní části, má horší stabilitu svislé osy, což může způsobit problémy při zvětšení pracovní plochy. Poslední nevýhodou je závislost na kvalitě podlahy, do níž je manipulátor umístěn. Při nerovnostech podlahy může docházet k ohybu vedené osy X a ke zvětšení tření na vedení, dále také k úplnému blokování pohybu.

Kvůli těmto nedostatkům bylo rozhodnuto o vytvoření další varianty uspořádání stroje.



Obrázek 4.1 První varianta realizace stroje

4.2 Druha varianta

Z pohledu na nedostatky předchozí varianty byl model opraven tak, aby se všechny eliminovaly. Konstrukce byla modifikována tak, že nyní je manipulátor montován k čelní straně tiskáren. Na obrázku 4.2 je vidět opravený model manipulátoru. Dva vodorovné profily tvoří čelní stranu tiskáren, kam musí být manipulátor montován. Oproti předchozí variantě bylo zde pootočeno vedení osy Y a vozík osy Z tak, aby se mohlo pohybovat dovnitř tiskáren.

Takovým způsobem lze řešit problémy způsobené umístěním manipulátoru na podlaze. Díky tomu, že je nyní montován k tiskárnám, konstrukce je nezávislá na kvalitě podlahy a závisí jenom na poloze tiskáren. Také není nutné s takovým uspořádáním používat profily využívané pro montáž osy X, čímž lze dosáhnout nižší ceny.

Hlavní nevýhodou takové konstrukce je nemožnost obsluhování tiskáren pro obě strany manipulátoru. Další nevýhody jsou: omezená výška odebíraného tisku, menší možná vzdálenost posuvu vozíku osy Z, komplikace s umístěním motorů os X a Y.

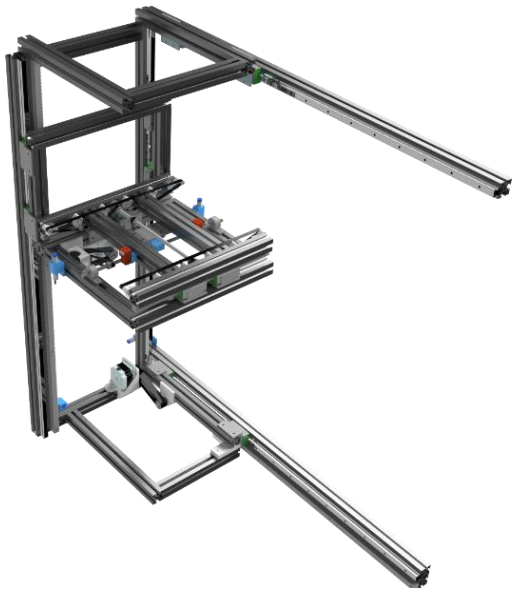
Kvůli daným nedostatkům nelze splnit požadavky manipulátoru, proto byla zvolena další varianta realizace stroje.

Obrázek 4.2 Druha varianta realizace stroje

4.3 Třetí varianta (finální)

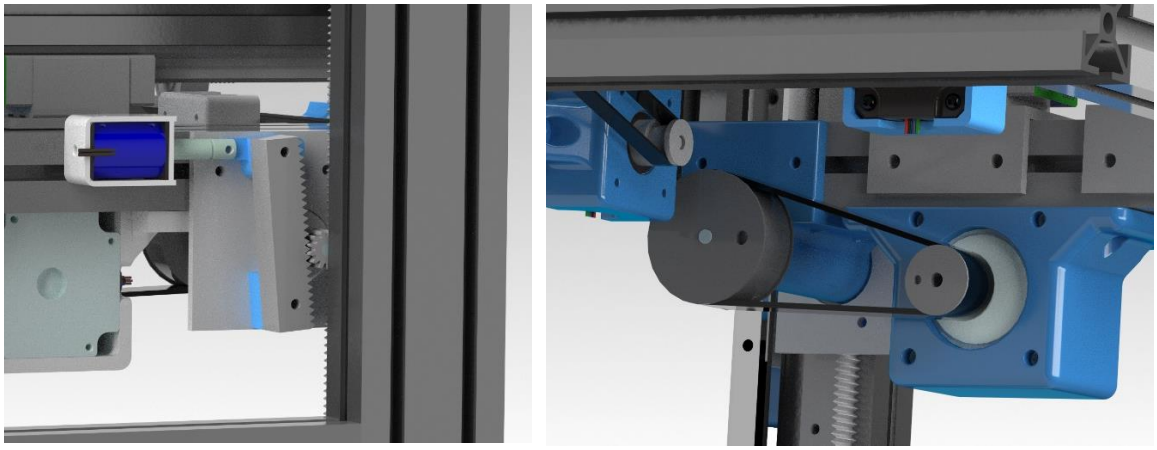
Třetí varianta je kombinací předchozích dvou. Manipulátor je zde montován k čelní straně tiskáren, ale uspořádaní osy Y a Z zůstává stejné jako u první varianty (viz Obrázek 4.3). Konstrukce má rozměry 1 × 1 m a je přizpůsobena pro

obsluhování 4 tiskáren z každé strany manipulátoru. V případě většího počtu tiskáren je možnost rozšíření o 1 m z každé strany. Během návrhu stroje bylo zvažováno, že minimální výška odběru musí být co nejmenší, aby měl manipulátor možnost odebírat tisk libovolného rozměru.



Obrázek 4.3 Finální varianta konstrukce manipulátoru

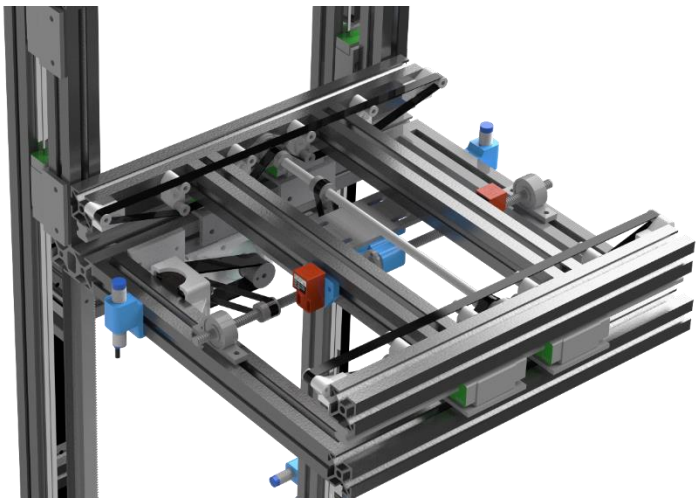
Pro splnění požadavků na minimální výšku odběru bylo nutné si rozmyslet, jak správně umístit motory os X a Y, aby neomezovaly pohyb. Řešením se stalo využití řemenových převodovek. Na obrázku 4.4b je vidět, jak je realizována převodovka osy Y. Motor je posunut dovnitř manipulátoru tak, aby nevadil pohybu. Ukotvení motoru je provedeno pomocí vytištěného plastového dílu.



a) b)

Obrázek 4.4 Brzdy (a) a převodovka (b) osy Y

Při vypnutí napájení motoru bude pohyb osou Y blokován pomocí mechanismu brzdění, který je realizován s využitím kusu ozubeného hřebenu, jenž bude pomocí pružiny blokovat otáčení ozubeného kola. Při zapnutí napájení motoru bude hřeben odtažen pomocí solenoidu, stejným způsobem se uvolní pohyb (viz Obrázek 4.4a).



Obrázek 4.5 Mechanismus odběru tiskové podložky

Na obrázku 4.5 je vidět podrobnější pohled na konstrukce osy Z. Mechanismus odběru podložky je realizován jako vozík, který se může pohybovat od nulové polohy 100 mm oběma směry, což umožňuje obsluhovat tiskárny z obou stran manipulátoru. Na vozíku jsou umístěny dva řemeny, z každé strany jeden, které slouží jako pásový dopravník. Až se vozík dostane k tiskárně, vysune se směrem dovnitř a pomocí pásového dopravníku odebere, nebo umístí podložku pro tisk.

Pohyb vozíku je zajištěn pomocí šroubů a krokového motoru Nema17. Limitní polohy jsou sledovány pomocí dvou indukčních snímačů umístěných po obou stranách manipulátoru. Přítomnost podložky na manipulátoru je detekována pomocí dvou kapacitních snímačů umístěných na vozíku.

5 NÁVRH SCHÉMATU ELEKTRICKÉHO ZAPOJENÍ

V dané kapitole je proveden návrh elektrického rozvodu pro manipulátor a návrh desek plošných spojů pro Arduino a Beaglebone Black. Návrh elektrického rozvodu byl proveden v programu WSCAD Suit X, který umožňuje plánovat systémy pro průmyslové automatizace. Desky byly navrženy v programu Autodesk Eagle.

5.1 Návrh elektrického rozvodu

Schémata elektrického obvodu jsou umístěna v přílohách 3 až 7. Rozvod je navržen pro využití motorů s optickými enkodéry a průmyslové drivery s řízením pomocí protokolu EtherCAT. Na první stránce je vidět napájecí a bezpečnostní obvod. Do stroje je přivedeno napětí 220 V. Napájení je vedeno přes hlavní vypínač, poté přes stykač, který bude plnit ochrannou funkci. Ochrana je realizována pomocí speciálního tlačítka EMERGENCY STOP a bezpečnostních koncových spínačů. Všechny spínače a tlačítka jsou zde sériově zapojena a při jejich stisknutí, nebo při nárazu vozíku na spínač dojde k rozpojení jeho napájení, tím se vypne napájení celého stroje. Napájecí obvod je rozdělen do tří větví: dvě mají 48 V a jsou pro napájení krokových motorů, jedna má 24 V a slouží k napájení elektroniky. Použité zdroje napájení jsou probrané v kapitole 3.3.4 Napájení.

Další dvě stránky vysvětlují propojení driverů a motorů mezi sebou, je zde nutné propojit výstupy driverů se vstupy motorů a výstupy z enkodéru připojit na vstupy driverů.

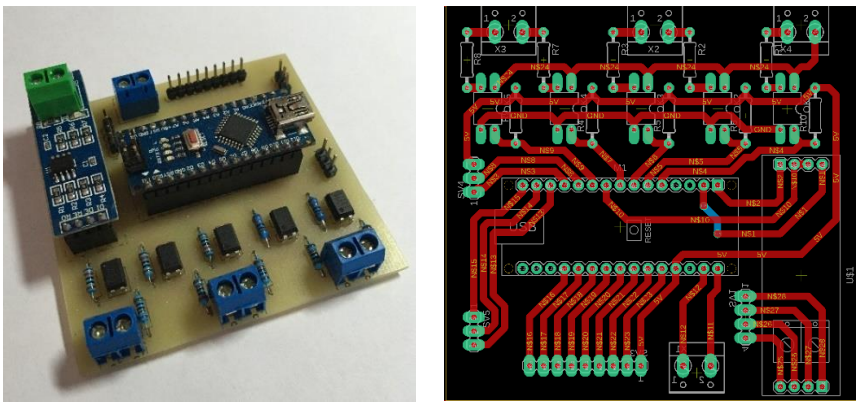
Na posledních dvou stránkách je ukázáno vzájemné propojení Beaglebone a Arduino a připojení periférie na ně. Řídicí přístroje jsou napájeny pomocí step-down modulů, které sníží napětí 24 V na nutných 5 V. Na Beaglebone je připojeno ještě jedno bezpečnostní tlačítko, které realizuje vypnutí napájení motoru softwarovým způsobem. Při stisknutí daného tlačítka bude vypnut signál Enable na drivery, čímž bude vypnuto napájení motoru.

Na Arduino jsou připojeny čtyři limitní snímače a dva krokové motory s řízením pomocí STEP/DIR signálů.

5.2 Deska plošných spojů pro Arduino

Pro jednoduchost připojení snímačů a modulu komunikace MAX-485 bylo rozhodnuto vytvořit pro Arduino desku plošných spojů. Hlavním úkolem této desky je převod napětí z výstupu snímačů, které je rovno napájecímu napětí na

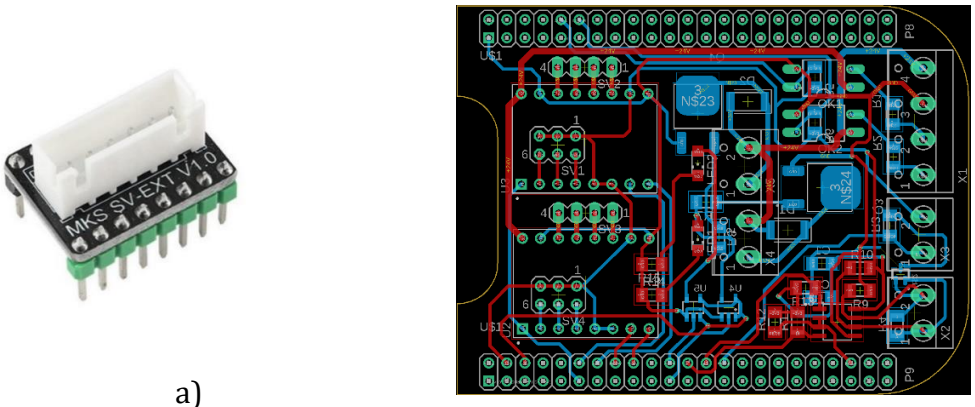
nutných 5 V pro Arduino. Daná funkce je realizovaná pomocí optočlenů PC817. Navržená deska umožňuje připojení až 5 snímačů a dvou krokových motorů. Schéma zapojení lze vidět v příloze č. 10, deska a její spoje jsou zobrazeny na obrázku 5.1.



Obrázek 5.1 Deska plošných spojů pro Arduino

5.3 Deska plošných spojů pro Beaglebone

Aby bylo možno zajistit potřebnou funkčnost, byla pro Beaglebone navržena deska plošných spojů, jež zahrnuje: podporu komunikace pomocí RS-485, možnost připojení dvou krokových motorů a driverů pro ně, podporu připojení dvou limitních snímačů a tlačítka Estop. Dále má dva výstupy, které jsou spínané pomocí tranzistorů. Schémata pro danou desku jsou umístěna v přílohách č. 8 a 9.



a) b)

Obrázek 5.2 Adapter pro připojeni driveru (a) a navrhnuta deska (b)

5.3.1 Drivery pro motory

Vzhledem k tomu, že pro stavbu prototypu stroje jsou použity krokové motory s řízením pomocí STEP/DIR signálů a možnosti připojení na konektor pro standardní drivery typu A4988 (viz Obrázek 5.2a), bylo by vhodné pro Beaglebone

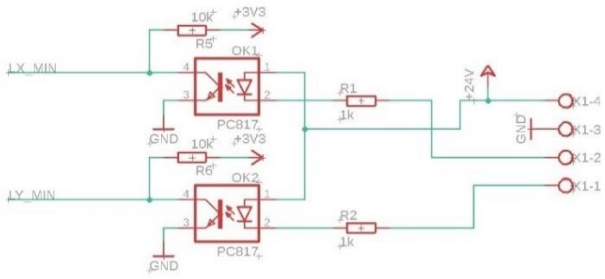
realizovat desku, která by mohla být použita nejen pro manipulátor, ale i pro jiné účely s využitím driverů A4988. Z tohoto hlediska byly umístěny na desce konektory pro drivery daného typu. K nim jsou přivedeny základní signály (STEP, DIR a Enable) a napájení. Pomocí propojek lze nastavit mikrokrokování.

5.3.2 Sběrnice RS-485

Pro realizaci komunikace pomocí protokolu RS-485 byl zvolen obvod MAX485ESA a použito bylo standardní zapojení. Pro ochranu proti elektrostatickému výboji (ESD) je použit obvod SP1001-02, který je realizován jako dvě TVS diody umístěné do jednoho pouzdra. Takovým způsobem je zajištěna ochrana proti výbojům do 15 kV.

5.3.3 Vstupy

Stejně jako u desky pro Arduino vstupy jsou galvanicky oddělené pomocí optočlenů PC817, což umožňuje napojit snímače na 24 V a zajistit bezpečnost pro vstupy Beaglebone.

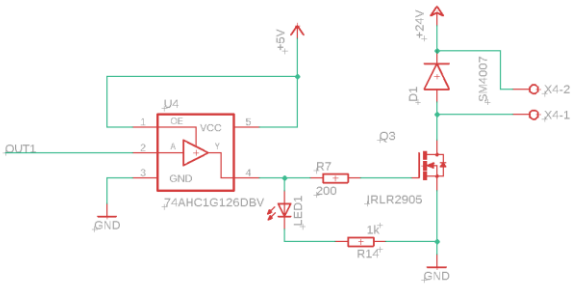


Obrázek 5.3 Schéma zapojeni pro vstupy

5.3.4 Výstupy

Výstupy jsou spínány pomocí tranzistorů IRLR2905. Maximální řídicí napětí daného tranzistoru je 16 V. Maximální proud činí 30 A. Aktivní výstup je indikován pomocí LED diody.

Výstupní proud z Beaglebone je zesilován pomocí bufferu 74AHC1G126DBV. Takovým způsobem lze zajistit bezpečnost před přetížením na výstupech mikropočítače.



Obrázek 5.4 Schéma zapojeni výstupu

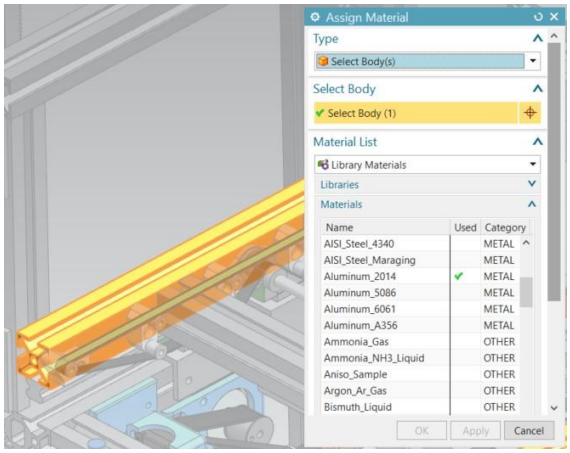
6 VIRTUALNI POHYBLIVY MODEL

Návrh pohyblivého virtuálního modelu bylo provedeno pomocí rozšíření Mechatronix Concept Designer pro Siemens NX (dále MCD). Pomocí daného softwaru je možné nastavit fyzické a mechanické vlastnosti modelu. MCD umožňuje propojení modelu s virtuálním nebo fyzickým PLC a takovým způsobem je možné provádět testování a nastavení softwaru pro reálný stroj.

6.1 Nastaveni modelu

Spustit MCD pro otevřený model je možné pomocí tlačítka *Mechatronix Concept Designer* v záložce *Application* programu Siemens NX. Po stisknutí se objeví všechny nutné funkce pro práce s MCD.

Prvním krokem je nutné zkontrolovat nastavení materiálu pro všechny díly modelu. Přirazení materiálu pro díl je možné provést pomocí funkce *Assign Materials*. Po vyvolání funkce se objeví podokno, ve kterém je nutné vybrat díl a materiál. Na obrázku 6.1 je uveden příklad přiřazení materiálu hliník pro stavebnicový profil.

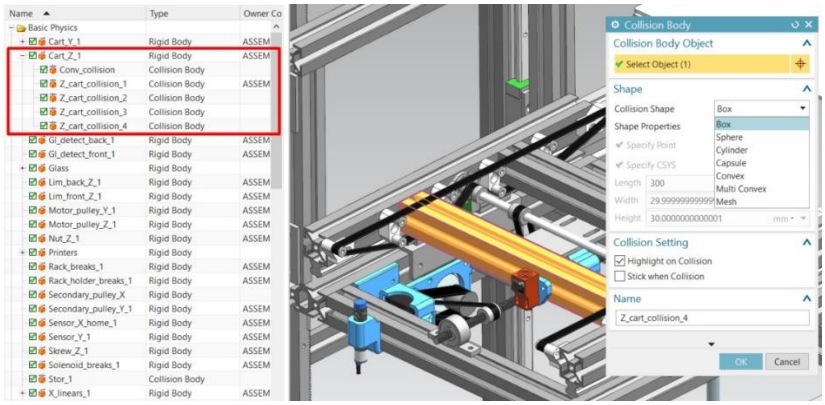


Obrázek 6.1 Nastaveni materiálu pro díl

Dalším krokem je nutné vytvořit tuhá a kolizní tělesa. Pro vytvoření tuhého tělesa slouží funkce *Rigid Body*. Ve vzniklém podokně je nutné vybrat všechny díly, které budou sloužit jako jediné těleso. Po vytvoření tělesa mu bude možné přiřadit vlastnosti jako hmotnost, momenty setrvačnosti, které na něj začnou působit určitými silami, a poté bude přidělena tělesu schopnost pohybu.

Kolizní tělesa slouží pro sledování dotyku. Pro jeho vytvoření slouží funkce *Collision Body.* Aby se dalo těleso nastavit jako kolizní, je nutné v podokně *Collision Body* vybrat součástku a její tvar. Tvar lze vybírat ze sedmi možností. V daném bodě je důležité brát v potaz, že čím komplikovanější bude zvolen tvar (např.

Mesh), tím obtížnější bude simulace pro počítač. Pro optimalizaci je vhodnější na jedno tuhé těleso zvolit několik jednoduchých kolizních. Tímto způsobem je například vytvořen vozík osy Z, který se skládá ze čtyř kolizních těles typu *Box* pro profily a jednoho typu *Mesh* pro pásový dopravník (viz Obrázek 6.2).

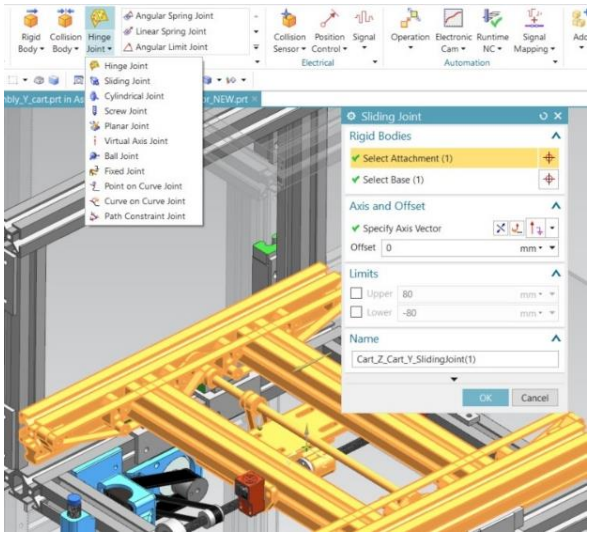


Obrázek 6.2 Tvorba kolizního tělesa

Po definici všech těles je možné přejít k nastavení vazeb. K tomu je v MCD využívána záložka *Joints*. Možné spoje v MCD jsou zobrazeny na obrázku 6.3. Během práce budou použity pouze vybrané z nich.

Prvním krokem pomocí funkce *Fixed Joint* byl vytvořen pevný spoj pro těleso, podle kterého se budou pohyby konat. Obvykle je to podlaha, v daném případě jsou to tiskárny. Pomocí daného spoje bude model pevně umístěn v prostoru.

Dále pomocí funkce *Sliding Joint* byly definovány lineární pohyby v osách X, Y a Z. Na obrázku 6.3 je ukázán příklad definice lineárního pohybu mezi vozíkem osy Y a vozíkem osy Z. Je zde nutné zvolit pohybující se těleso, podle něhož bude konán pohyb a směr.



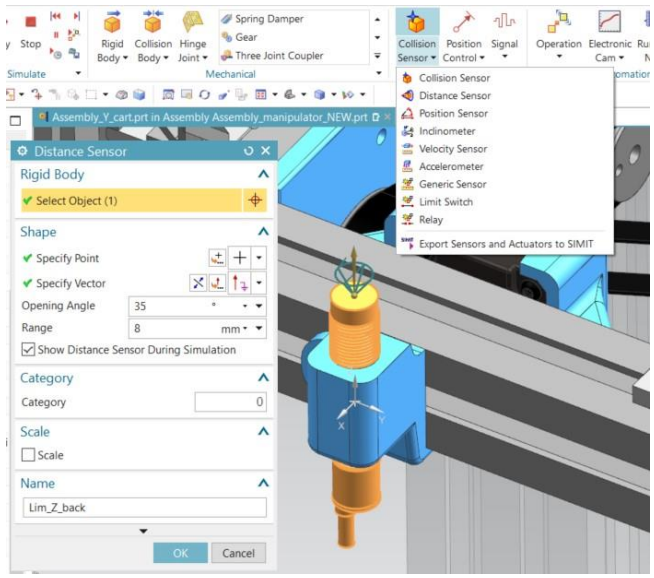
Obrázek 6.3 Nastaveni lineárního pohybu

Dalším typem spoje použitým v daném modelu je spoj otočení. Tento typ spojení je použit pro všechny řemenice a ložiska šroubu osy Z. Nastavení takového typu spoje se provádí pomocí funkce *Hinge Joint*.

Poslední využitou vazbou je šroub (*Skrew Joint*). Daný spoj je použit pro pohyb osy Z. Nastavení zde probíhá stejně jako pro *Hinge Joint*, jediným rozdílem je parametr *Pitch*, kde je třeba zadat stoupání šroubu, což činí 4 mm.

Po definici všech vazeb je možné určit převodovky. V modelu jsou použity dva typy převodovek: řemenová a převod mezi ozubeným kolem a ozubeným hřebenem. Pro vytvoření převodu slouží v MCD záložka *Couplers.* Vytvořit potřebné převodovky je možné pomocí příkazů *Gear* a *Rack and Pinion.* Po vyvolání funkce je nutné vybrat spoje, mezi kterými bude převod vytvořen, a zvolit převodový poměr.

V daném bodě jsou definovány všechny mechanické prvky a vazby mezi nimi a lze přejít k součástkám elektrickým, jako jsou snímače, motory a dopravníky.



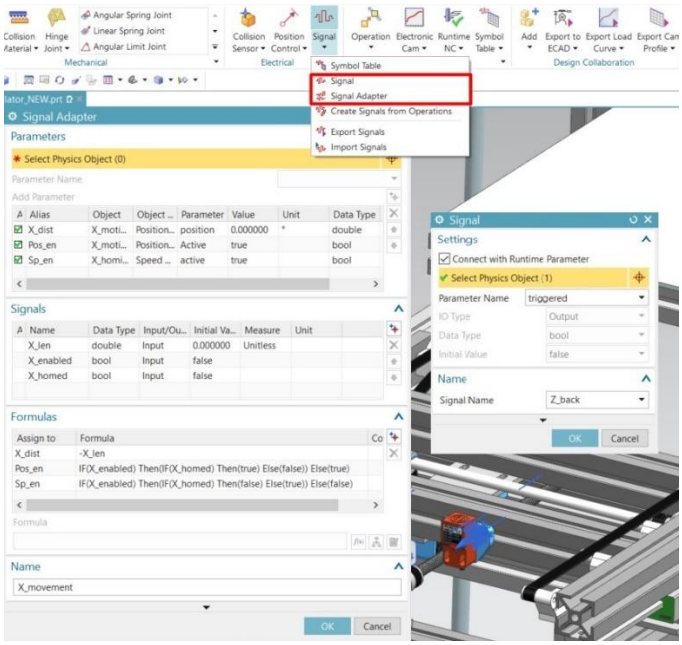
Obrázek 6.4 Nastaveni snímačů

Na obrázku 6.4 jsou vidět všechny snímače, které je možné použít v MCD. Z hlediska toho, že u manipulátoru jsou použity jenom kapacitní a indukční limitní snímače, proto jsou v simulaci všechny nastaveny jako *Distance Sensor*. Pro simulace úhlu a vzdálenosti byly detekce nastaveny na 35° a 8 mm.

Dalším krokem je vytvoření dopravníku. U manipulátoru jsou dopravníky použity na vozíku osy Z, a to v každé tiskárně. Pro nastavení dopravníku slouží funkce *Transport Surface*. Dále je nutné zvolit plochu kolizního tělesa, která bude sloužit jako dopravník a vektor pohybu. Pro všechny dopravníky byla zvolena rychlost pohybu 100 mm/s.

Nyní zbývá pouze nastavit řízení motorů. Pro každý z nich budou definovány dva typy řízení: *Speed Control* a *Position Control*. Tyto typy jsou nezbytné k tomu, aby bylo možné realizovat pohyb do nulové polohy každé osy, a poté řízení na vzdálenost. Podrobnější popis chování bude vysvětlen v dalších kapitolách. Pro motory lze nastavit maximální rychlost a zrychlení, tyto hodnoty budou vypočteny a otestovány v další kapitole. Nyní nám postačí defaultní hodnoty.

Posledním krokem je nastavování signálů a signál adapterů. Signál je možné vytvořit pro téměř všechny parametry v MCD. Příklad vytvoření signálu bude ukázán pro výstup z limitního snímače osy Z. Signály budou vytvořeny pro všechny hodnoty, např.: výstupy limitních snímačů, hodnota vzdálenosti pohybu pro motory atd.



Obrázek 6.5 Tvorba signálu a signál adapteru

V případech, kdy je pro jeden objekt nutno využít několik signálů, bude vhodnější využít *Signal Adapter*. Signály adaptery jsou tady použity pro řízení motorů třemi signály (vzdálenost pohybu, signál Enable a signál pro houmování). Pomocí signálu adapteru je možné zvolit jeden nebo několik parametrů, které budou tvořit signály, na jejichž základě budou změny probíhat. Na obrázku 6.5 je ukázán příklad signálu adapteru pro motor osy X.

V daném bodě je model připraven pro simulace a je možné přejít k nastavení komunikace s řídicím softwarem.

6.2 Nastaveni komunikace

MCD podporuje několik způsobů komunikace se softwarem nebo fyzickým zařízením, a to jsou:

- OPC DA
- OPC UA (v novějších verzích Siemens NX)
- SHM
- Matlab
- PLCSim Advanced
- TCP
- UDP
- Profinet

Prvním krokem bylo rozhodnuto se pokusit rozjet komunikace mezi MCD a programem Codesys pomocí určitého průmyslového protokolu. Na začátku byl zvolen protokol Profinet, který je podporován jak Codesysem, tak MCD. Po nastavení několika proměnných pro testování a komunikace pomocí Profinet v Codesys, bylo zjištěno, že není způsob, aby bylo možno rozjet komunikace pomocí daného protokolu bez využití fyzického zařízení. Simulace Profinetu na počítací pomocí Codesys není možná. Z tohoto hlediska bylo rozhodnuto využít jiný způsob komunikace.

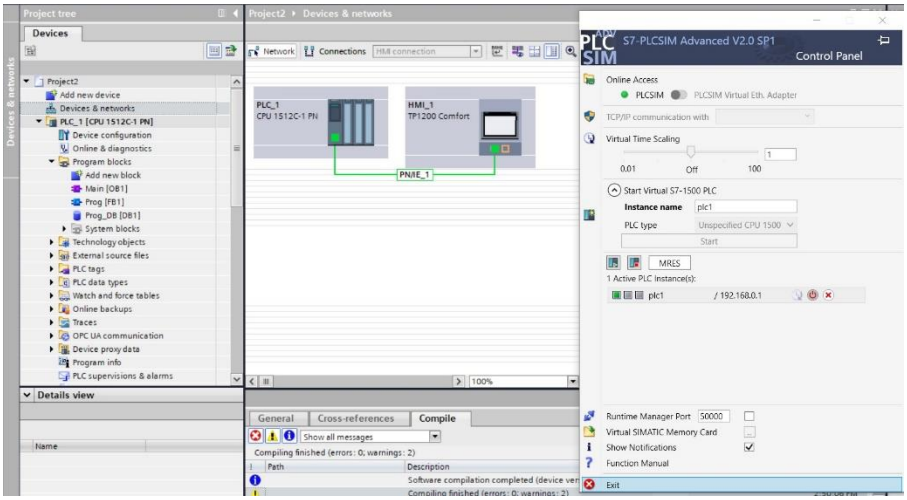
Dalším krokem bylo rozhodnuto vyzkoušet komunikace pomocí OPC DA. OPC DA server pro Codesys není zdarma a je nastavován pomocí externího programu OPC Configurator. Pro testování byla použita trial verze daného programu. Pro nastavení OPC DA serveru je nutné spustit program a v něm zadat stejný název PLC, jako je název v Codesys.

Dalším krokem je nutné připojit MCD na OPC server. V daném kroku vznikl problém. MCD nedetekoval OPC DA server. Při výběru komunikace pomocí OPC DA se server neobjevoval v seznamu možných připojení. Při pokusech připojit se na adresu serveru (CoDeSys.OPC.DA) ručně MCD ukázal chybu „Cannot connect to this OPC server“. Pro zjištění, je-li problém na straně Codesys nebo MCD, bylo rozhodnuto vytvořit ještě jeden server pomocí programu testování OPC sítě. Pro vytvoření serveru byl použit program KEPServerEX, který utvoří OPC server s několika signály různých datových typů. Nový server se také neobjevoval v seznamu zapojení. Bylo rozhodnuto, že problém je v MCD. Následně bylo zjištěno, že takové chyby mohou vznikat, když nainstalovaná verze OPC Core Components není podporována MCD. Proto se zkusilo nainstalovat jiné verze Core Components. Po instalaci verzi 3.00.108 začal MCD detekovat oba vytvořené servery. Takovým způsobem se dalo nakonfigurovat komunikace a propojit signály mezi Codesysem a MCD.

Po detailnějším prozkoušení zapojení se ukázalo, že je to nestabilní. Během práce bylo velké zpoždění a často se ztrácelo připojení mezi MCD a serverem. Při změně proměnných v Codesys bylo nutné smazat všechny spojené signály v MCD a propojit je znovu, jinak vznikala chyba „Failed to acces tag“ pro všechny

definované proměnné. Po jedné z takových oprav vznikla chyba „Memory acces violation“ při libovolné manipulaci s nastavenými signály. Daný problém se řešil reinstalací Siemens NX. Kvůli problémům bylo rozhodnuto tento typ zapojení nevyužívat.

Dalším krokem se aplikoval software od Siemensu, který je přizpůsoben pro práci s MCD. Daným softwarem je TIA Portal a PLCSIM Advanced. Takové řešení se ukázalo jako úspěšné.



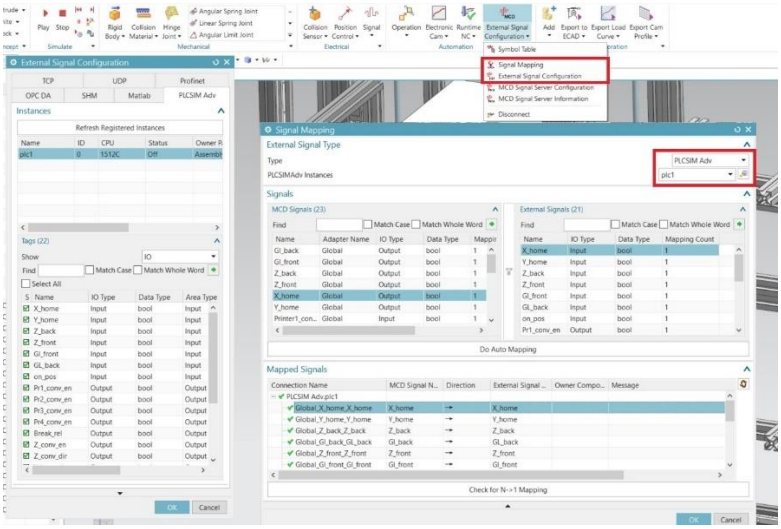
Obrázek 6.6 Nastaveni TIA Portalu a PLCSIM Advanced

Pro realizace takového typu zapojení je nejprve potřeba nastavit program PLCSIM Advanced. Aby bylo možné spustit virtuální PLC, je nutné v sekci *Start Virtual S7-1500 PLC* zadat jméno budoucího PLC a stisknout tlačítko *Start*. Vytvořený PLC se objeví v seznamu *Active PLC Instance (s), kde* je ukázána adresa PLC a tlačítka pro jeho ovládání. Je nutné zkontrolovat, zda je vytvořený PLC zapnutý, a tím samým způsobem nastavení PLCSIM Advanced ukončit.

Dále je možné nastavit TIA Portál v předchozím bodě PLC. Pro simulaci lze využít libovolný PLC ze série S7-1500. Byl zvolen procesor CPU 1512C-1 PN se stejnou IP adresou jako vytvořený v PLCSIM Advanced. K němu je připojen HMI panel TP1200 Comfort pro ovládání programu. Na obrázku 6.13 je vidět nastavení použité v PLCSIM Advanced a zvolený hardware v TIA Portal. Po nastavení PLC je potřeba vytvořit globální proměnné v záložce PLC tags, zde budou proměnné využívané pro komunikace s MCD. Po definici proměnných je možné nahrát program do PLC a vrátit se do MCD, aby se mohly spojit signály.

Příklad propojení signálů je ukázán na obrázku 6.14. Na začátku je nutné nakonfigurovat vnější signály pomocí funkce *External Signal Configuration*. Typ propojení je *PLCSIM Adv*. Po výběru typu komunikace se zobrazí seznam možných připojení. Je zde vidět vytvořený drive PLC s názvem plc1. Pro zobrazení signálů od PLC je nutné ho vybrat v seznamu a tím samým způsobem se v sekci zobrazí *Tags* a

všechny globální proměnné z PLC. Jestli se signály neobjeví, je nutné stisknout tlačítko *Refresh Registred Instances*, a tím samým způsobem se načte MCD ještě jednou. Ze seznamu *Tags* je potřeba vybrat signály, které budou spojeny s vnitřními signály MCD. Nastavení se potvrdí tlačítkem *OK*.



Obrázek 6.7 Propojovaní signálu

Dalším krokem je propojení vnějších signálů se signály vnitřními. Lze to udělat pomocí funkce *Signal Mapping*. Po vyvolání funkce se objeví okno, ve kterém je nejprve nutné vybrat typ externího signálů a PLC nastavený v předchozím bodě. Dále jsou vidět tři sekce: *MCD Signals* (vnitřní signály), *External Signals* (signály z PLC) a *Mapped signals* (propojené signály). Propojit signály je možné dvěma způsoby: automaticky a ručně. Automapování lze použít tehdy, když mají vnitřní a vnější signály stejné názvy. Tehdy budou stisknutím tlačítka *Do Auto Mapping* spojeny všechny vnitřní signály, které mají pár ve vnějších signálech. Pro ruční propojování je potřeba vybrat vnitřní signál ze sekce *MCD Signals*, poté zvolit vnější signál ze sekce *External Signals* a stisknout tlačítko *Map Signals*. Na obrázku 6.14 je vidět příklad propojení signálu X_home. Takovým způsobem jsou namapovány všechny signály.

Nyní je model připraven pro simulace a je spojen s virtuálním PLC. Dalším krokem je vytvoření programu pro jeho řízení.

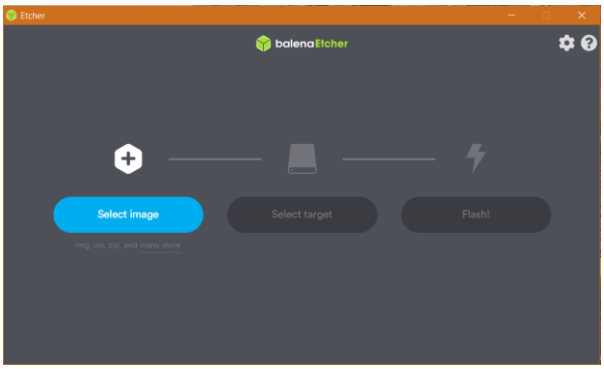
7 NAVRH SOFTWARU

7.1 Nastaveni Machinekit

Jako operační systém pro Beaglebone byl vybrán Machinekit. Tento systém přizpůsobený pro řízení strojů podporuje řízení v reálném čase a má dobrou flexibilitu pro tvorbu vlastních řešení.

7.1.1 Instalace Machinekitu

Prvním krokem je nutné nainstalovat operační systém do eMMC paměti Beaglebone Black. Stáhnout poslední verze Machinekitu lze ze stránky elinux.org. Po stažení bude nutné vypálit obraz operačního systému do paměťové karty. Pro toto byl použit program Balena Etcher, kde je nutné zvolit stažený operační systém, paměťovou kartu a stisknout tlačítko *Flash* (viz obrázek 7.1).



Obrázek 7.1 Program Balena Etcher

Aby z připravené paměťové karty byl systém nainstalován do eMMC paměti, je nutné provést jedno nastavení. Na paměťové kartě je potřeba modifikovat soubor */boot/uEnv.txt* tak, aby byl odstraněn poslední řádek.

```
  cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

Nyní je možné do vypnuté desky Beaglebone Black vložit paměťovou kartu a připojit napájení. Po připojení napájení se automaticky spustí instalace operačního systému. Indikace instalace je realizována postupným blikáním ledek a může trvat až 40 minut. Po skončení instalace všechny ledky zhasnou.

Machinekit bude řízen pomocí Python skriptu, jehož funkčnost bude vysvětlena v další kapitole. Na začátku je nutné vytvořit konfigurační soubory, na jejichž základě bude určena funkčnost Machinekitu.

7.1.2 Konfigurace GPIO

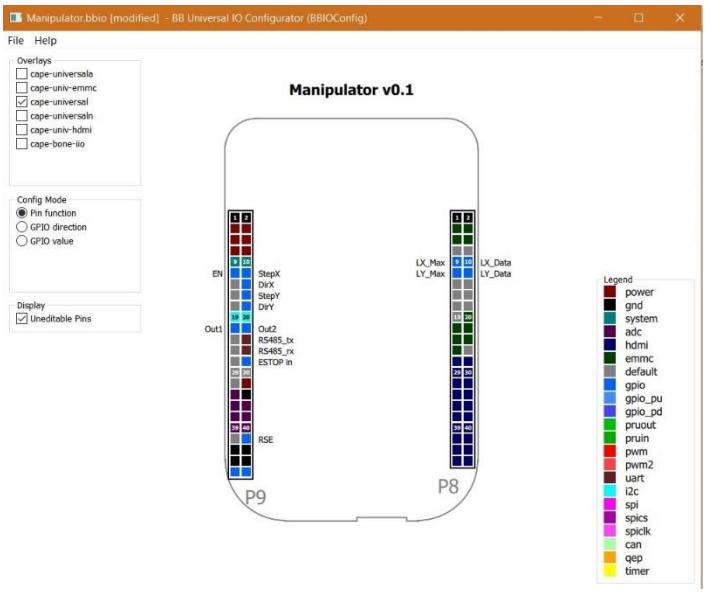
Prvním krokem je nutné nastavit vstupy a výstupy Beaglebone. Lze to udělat pomocí příkazového řádku nebo *.bbio* souboru. V daném případě budou vstupy a výstupy nastavovány automaticky při spouštění Python skriptu, proto je vhodnější využít *.bbio* soubor. Nastavení v souboru jsou pro každý využitý pin Beaglebone zapsána postupně. Zápis je tady prováděn ve tvaru „číslo pinu“, „režim“.

Například:

```
P8_10 in
```

Pin P8_10 je zde nastaven na vstup. Piny je možné nastavit na režim vstupu (*in*) a výstupu (*out*). Některé piny lze nastavit pro komunikace (*uart, spi,* atd.). Jestli je pin nastaven na výstup, je možné nastavit jeho logickou úroveň (*high, low*). Možnosti nastavení každého pinu jsou uvedeny v datasheetu.

Pro jednoduchost nastavení lze využít program BBIOConfig, jehož uživatelské prostředí je ukázáno na obrázku 7.2



Obrázek 7.2 Program BBIOConfig

Prvním krokem je nutné vybrat ze sekce *Overlays* alespoň jednu skupinu pinů k využití. Některé piny jsou používané pro interní zařízení Beaglebone (paměť, HDMI výstup), proto je nutné se vyhnout jejich použití, nebo vypnout zařízení v nastaveních Machinekitu. Pro možnost zvolení pouze volných pinů jsou zde vytvořeny skupiny podle zařízení jejich využívajícího. Například skupina *cape-univ-emmc* využívá piny pro eMMC paměť, skupina *cape-univ-hdmi* využívá piny pro HDMI výstup. Pro realizace byla zvolena skupina *cape-universal*, která obsahuje jenom volné piny, nevyužité žádným zařízením.

Dalším krokem je možné nastavit všechny potřebné piny na požadovaný režim práce. Možnosti zde jsou: *gpio, gpio pullup, gpio pulldown*, nebo komunikace *uart, spi, i2c*. Dále je nutné se v sekci *Config Mode* přepnout do *GPIO direction* modu. Zde je možné v předchozím bodě nastavené piny GPIO nastavit jako vstupy nebo výstupy. Potom je možné pomocí modu *GPIO value* zvolit jejich logickou úroveň. Po dokončení nastavení příkazem *Save* budou nastavení uložena ve formátu *.bbio* souboru.

7.1.3 Konfigurační soubor Machinekit

Na základě dokumentace Machinekit a standardního konfiguračního souboru pod názvem CRAMPS byl vytvořen konfigurační soubor pro manipulátor. Tento soubor musí obsahovat základní konfigurace pro Machinekit a nastavení pro dvě osy.

Prvním krokem je konfigurována sekce PRUCONF. Daná sekce odpovídá za generaci vysokofrekvenčních signálů, jako jsou PWM a STEP. Nastavení dané sekce lze nechat defaultně, potřeba je jenom změnit počet nutných PWM a STEP výstupu. V tomto případě jsou používány jenom dva STEP výstupy, které jsou nastaveny jako `num_stepgens=2`.

Další sekce je EMC, kde jsou uvedeny základní informace pro stroj, jako jsou název a verze. Lze nastavit debug úroveň, která bude určovat, kolik hlášení se vepíše do terminálu.

V sekci DISPLAY jsou nastavení pro uživatelské rozhraní Machinekitu. Lze zde zvolit požadované prostředí a jeho nastavení. Manipulátor nebude ovládán pomocí standardních uživatelských prostředí, proto byl parametr DISPLAY nastaven jako *dummy*, čímž bude uživatelské prostředí vypnuto.

Dále jsou sekce TASK, RS274NGC a EMCMOT, které odpovídají za nastavení motion controlleru a překladačů g-codu. Dané nastavení je vhodné nechat defaultní.

Do sekce HAL je potřebné zadat název (*.hal*) souboru vytvořeného pro daný stroj. Protože nebude použito uživatelské prostředí, není nutné nastavovat další parametry.

Další sekce je TRAJ. Do ní je nutné zadat počet os stroje, jejich souřadnice a maximální rychlost pohybu v nich. Je nutné nastavit jednotky lineárního (*mm, inch*) a rotačního (*deg, rad*) pohybu.

Posledním krokem jsou nastavení každé z os. Pro každou z nich je vytvořena zvláštní sekce s názvem AXIS_<NUM>, kde NUM je číslo osy. V daných sekcích je nutné nastavit typ pohybu (*linear, angular*), maximální rychlost, zrychlení, dolní a horní limit vzdálenosti pohybu a další parametry. Celkový seznam možných nastavení je uveden v dokumentaci Machinekit.

Vytvořený konfigurační soubor je umístěn v přílohách pod názvem Manipulator.ini.

7.1.4 HAL soubor

HAL (Hardware Abstraction Later) je soubor, který slouží jako vrstva mezi hardwarem a softwarem. Tento soubor byl vytvořen pomocí návodu v dokumentaci a standardní konfigurace CRAMPS.

Prvním krokem pomocí příkazu `loadrt` jsou do souboru nahrány všechny jeho nezbytnosti pro funkčnost jako například motion controler a nízkoúrovňové drivery. Přičemž při jejich konfiguraci jsou použity hodnoty z *.ini* souboru vytvořeného v předchozím bodě.

Dalším krokem jsou pomocí příkazu `addf` funkce přidávány do podprocesů. Příkaz je používán ve tvaru: `addf <funkce> <podproces>`. Existují dvě možnosti podprocesů. Pro funkce, které potřebují co nejrychlejší odpověď, je používán *base-thread*. Pro ostatní je používán *servo-thread*.

Dále se provádí nastavení pro každou osu. Je zde důležité propojit signály *enable*, signál odpovídající za předávání souřadnic pro pohyb, signál zpětné vazby polohy osy, jsou tady nastavovány piny výstupů STEP a DIR. Vytvořit nový signál je možné pomocí příkazu `newsig`, pro nastavení jeho hodnoty lze využít příkaz `sets`. Propojovat signály a piny je možné pomocí příkazu net. Například zde:

```
net emcmot.00.enable => hpg.stepgen.00.enable
```

Signál *enable* motoru 0 (X) je propojen s pinem odpovídajícím za aktivace výstupního signálu STEP. Nastavení parametrů rychlosti, zrychlení atd. jsou načítané z inicializačního souboru. Zbývá zde pouze nastavit piny pro výstupy STEP a DIR:

```
setp hpg.stepgen.00.steppin         912
setp hpg.stepgen.00.dirpin          914
```

Takovým způsobem je signál STEP nastaven na výstup P9_12 a signál DIR na výstup P9_14 Beaglebone.

Po nastavení os je nutné nakonfigurovat vstupy a výstupy. Jako výstup je nastaven signál EN (*enable*) pro drivery motorů. Vstupy jsou nutné pro tlačítko Estop a pro všechny limitní snímače. Příklad pro snímač nulové polohy osy X:

```
newsig limit-x-min bit
net limit-x-min <= bb_gpio.p8.in-10
net limit-x-min => axis.0.home-sw-in
```

Na začátku je vytvořen signál pro detekce nulové polohy osy X, dále je do signálu načítán stav vstupu P8_10 Beaglebone. Posledním krokem je hodnota signálu předávaná do Machinekitu jako detekce nulové polohy osy 0.

Pro různé snímače může nastat situace, kdy je nutné využívat obrácenu hodnotu od hodnoty vstupné. Toto lze udělat pomocí příkazu:

```
setp bb_gpio.p8.in-10.invert 1
```

Po konfiguraci vstupů a výstupů je *.hal* soubor vytvořen. Celkový jeho kód lze najít v přílohách pod názvem Manipulator.hal.

Posledním využitým souborem je soubor setup.sh, který odpovídá za konfigurace PRU a nastavení pinů. Lze ho použít stejně jako pro standardní konfigurace.

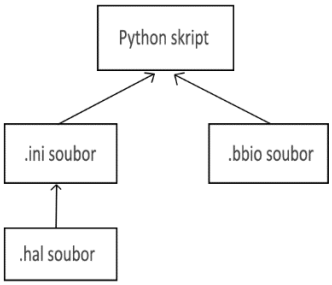
Po vytvoření daných konfiguračních souborů je možné přejít k vytvoření Python skriptu.

7.2 Skript pro Beaglebone

Nainstalovaný na Beaglebone Machinekit je ovládán pomocí Python skriptu. Prvním krokem po spuštění skriptu bude čekání na připojení klienta. Komunikace je zde realizovaná pomocí TCP/IP s využitím knihovny socket. Celkový kód pro vytvoření komunikace:

```
import socket
HOST = ''
PORT = 7000
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind((HOST, PORT))
sock.listen(1)
conn, addr = sock.accept()
```

Daný kód na začátku naimportuje knihovnu socket. Další dva řádky určují adresu a port budoucího serveru, což je zde localhost a port 7000. Dále je vytvořen socket a pomocí metody *bind* nastaven na práci se zvolenou adresou a portem. Metodou listen je dále socket spouštěn a je nastaven maximální počet připojení na 1. Posledním příkazem bude přijato připojení klienta. Funkce vrací socket, který bude použit pro komunikace a jeho adresu. Po rozjetí komunikace skript spustí Machinekit s využitím konfiguračních souborů vytvořených v předchozím bodě. Celková struktura programu je vidět na obrázku 7.2



Obrázek 7.3 Struktura programu pro Beaglebone

Pro načtení hodnot a odeslání příkazu do Machinekitu jsou používané příkazy:

```
manip = linuxcnc.stat()
command = linuxcnc.command()
manip.poll()
```

Metoda *poll()* načítá aktuální hodnoty z Machinekitu. Daný příkaz je vhodný pro použití před každým načítáním hodnoty.

Dále je nutné manipulátor převést do stavu zapnuto a vynulovat stav bezpečnostního tlačítka Estop. Toto je provedeno pomocí příkazů:

```
command.state(linuxcnc.STATE_ESTOP_RESET)
command.state(linuxcnc.STATE_ON)
```

Po provedení daných příkazů je Machinekit zprovozněn. Dalším krokem se manipulátor vrací do nulové polohy. Po vynulování klienta je odesláno potvrzení o zprovoznění manipulátoru. Program nyní přejde do hlavního cyklu, kde čeká na příkaz od uživatele. Před každým pohybem je kontrolován stav manipulátoru (tlačítko Estop, chyby). V případě, že nastane chyba, bude do klienta odesláno chybové hlášení.

Příkazy dostává program v bytovém tvaru a po dostání příkazu ho musí nejprve rozbalit. Pro rozbalení je použit příkaz *unpack()* z knihovny *struct*.

```
unpack = struct.Struct('f f I')
data = conn.recv(unpack.size)
data = unpack.unpack(data)
coord = 'G0 X{0:f} Y{1:f}'.format(data[0], data[1])
```

Postup vytvoření a odeslání balíčku ze strany klienta je vysvětlen v kapitole 7.3.2 Komunikace. Ze strany serveru je nutné provést stejný postup v opačném pořadí. Prvním příkazem jsou určeny datové typy prvků v balíčku a musí být nastavené stejné jako pro klienta. Pomocí dalších dvou příkazů dostane program balík od klienta a rozbalí ho. Posledním krokem je vytvořen textový příkaz pro Machinekit ve tvaru g-codu.

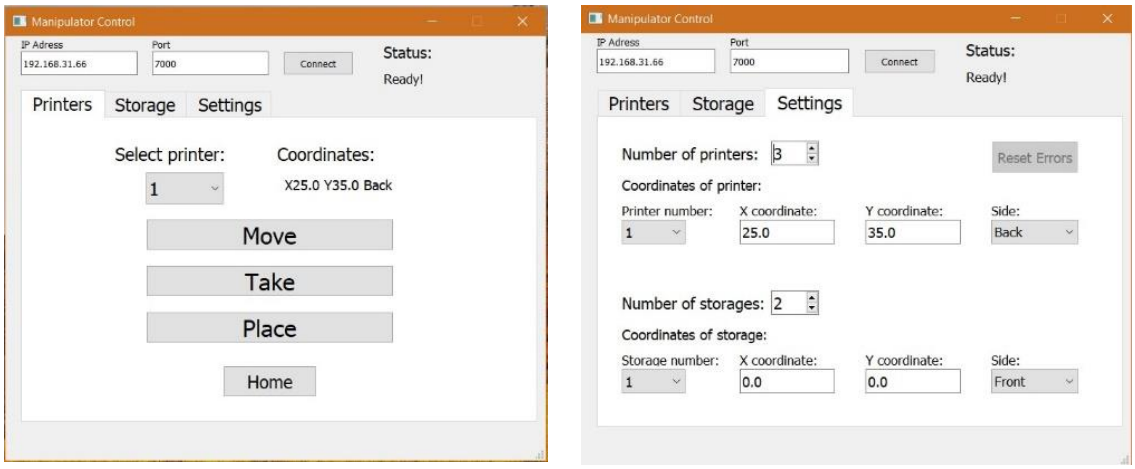
Dále je vytvořený řetězec odeslán do Machinekitu pomocí příkazu `command.mdi(coord)`. Po odeslání program čeká, dokud Machinekit nezmění svůj stav na *inpos*, což bude znamenat, že pohyb v osách X a Y je dokončen. V dalším kroku je odeslán příkaz do Arduino, na jehož základě bude provedena manipulace s tiskovou podložkou. Po dokončení činnosti vrací Arduino potvrzení, čímž bude skript převeden do stavu čekání na další příkaz.

Celkový kód programu je umístěn v přílohách.

7.3 Klient pro ovládaní z počítačů

7.3.1 Uživatelské rozhráni

Pro ovládání manipulátoru pomocí počítačů byl vytvořen klient, pomocí kterého lze nastavit počet tiskáren, skladu a souřadnice každé z nich.



Obrázek 7.4 Uživatelské rozhráni

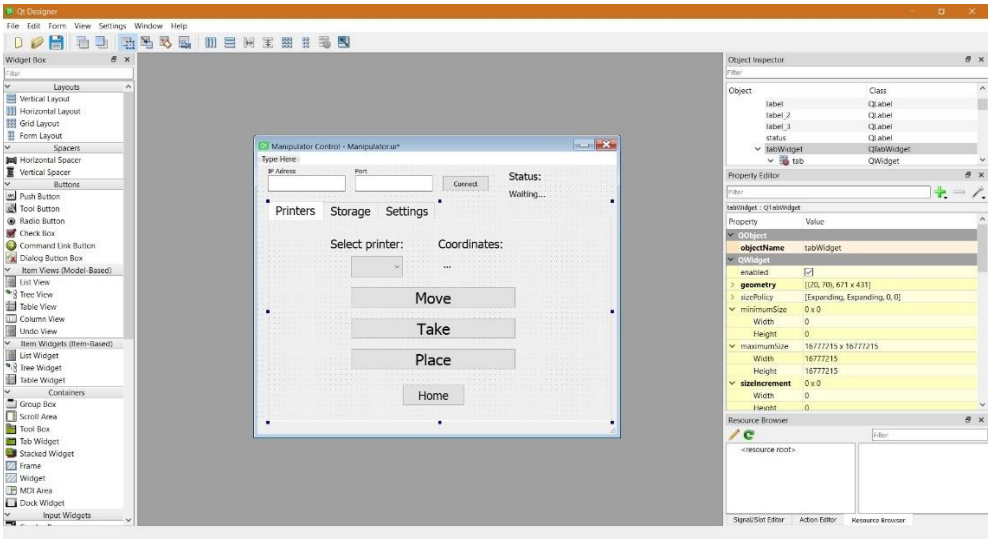
Grafické uživatelské rozhraní bylo vytvořeno pomocí programu Qt designer, který je součástí balíčku PyQt5. Nainstalovat ho je možné pomocí příkazového řádku příkazem: `pip install PyQt5`. Po spuštění programu se objeví jeho okno a navrhne možnosti vytvoření nového projektu. Pro dané účely byl založen nový projekt typu *Main Window*. Po stisknutí tlačítka OK bude zobrazen grafický editor a okno budoucího rozhraní (viz obrázek 7.2). V levé části okna je zobrazen seznam se všemi možnými grafickými součástkami. Pro přidání součástky je nutné ji přetáhnout ze seznamu do místa, kam musí být umístěna. V pravé části okna je umístěn seznam použitých součástek (*Object Inspector*) a nastavení pro vybranou součástku (*Property Editor*).

Prvním krokem pro okno byl nastaven statický rozměr. Toho lze dosáhnout nastavením stejného rozměru do pole *minimumSize* a *maximumSize* v *Property Editoru*. Rozměr byl nastaven na 700 × 550. Dalším krokem je umístění všech potřebných prvků. Pro nastavení IP adresy, portu a souřadnic tiskáren jsou použity prvky typu *Line Edit*. Záložky lze realizovat pomocí komponentu *Tab Widget.* Výběr tiskáren a skladu je realizován pomocí prvku *Combo Box*. Všechny další prvky jsou tlačítka (*Push Button*) a nadpisy (*Label*). Po přidání všech komponent je vhodné nastavit název pro každý prvek, aby byl jednoduchý k orientování. To lze pomocí *Property Editoru* a příkazu *objektName*.

Po nastavení všech nutných parametrů v *Property Editoru* je nutné projekt uložit a převést do Python kódu. Toto lze udělat s využitím příkazového řádku příkazem:

```
>> pyuic5 project.ui -o project.py
```

Kde *project.ui* je projekt vytvořený pomoci Qt designeru a *project.py*, soubor, do něhož bude výsledek umístěn. Výsledkem vykonání příkazu je Python soubor, kde se po spuštění zobrazí okno s uživatelským rozhraním, ale bez funkčnosti.



Obrázek 7.5 Program Qt designer

Aby nebyla část kódu zajišťující funkčnost klienta ovlivňována v případě změny uživatelského rozhraní, bude vytvořena jako zvláštní Python soubor. Do daného souboru je potřebné naimportovat komponenty nutné pro práci s Qt a soubor s vytvořeným v předchozím bodě rozhraním:

```
from PyQt5 import QtWidgets
import ManipGUI
```

Pro spojení programu s uživatelským rozhraním bude vytvořena nová třída, která bude mít možnost využívat všechny jeho funkce. Definice třídy:

```
class ManipulatorApp(QtWidgets.QMainWindow,
ManipGUI.Ui_ManipulatorControl)
```

Dalším krokem je inicializace třídy ve funkci *main*. To je realizováno pomocí kódu:

```
app = QtWidgets.QApplication(sys.argv)
window = ManipulatorApp()
window.show()
app.exec_()
```

Prvním krokem je zde vytvořen nový prvek QApplication, poté objekt třídy z předchozího bodu. Dalším krokem je spuštěno uživatelské rozhraní a program. Tím je inicializace dokončena a lze přejít k programování funkčnosti klienta.

Interakce uživatele s prostředím je sledována pomocí třídy ManipulatorApp. Dále je uveden příklad sledování zvolené tiskárny pro zobrazování jejích souřadnic:

```
  self.printer_sel_2.currentIndexChanged.connect(self.show_
coords)
```

Smysl daného příkazu je v tom, že jestli v poli s názvem *printer_sel_2* bude změněno číslo (*currentIndexChanged*), bude vyvolána funkce *show_coords*, která zobrazí souřadnice tiskárny pod daným číslem. Takovým způsobem lze definovat děje pro všechny prvky. Celkový kód lze vidět v přílohách.

7.3.2 Komunikace

Pro komunikace se serverem umístěným na Beaglebone je nutné realizovat klienta. Pro jeho realizace je použita knihovna socket. Stejné jako u programu je nutné pro Beaglebone na začátku vytvořit socket příkazem

```
  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Dále je ale postup rozlišný. Pro připojení na server je nutné zadat jeho adresu a port. Tyto hodnoty jsou odečítány s poli uživatelského prostředí s odpovídajícími názvy. Připojit se na server je možné pomocí příkazu:

```
  sock.connect((host, int(port)))
```

Z hlediska toho, že hodnoty s poli jsou odečítány ve formátu string, ale požadovaný formát pro port je int, je zde realizován převod mezi datovými typy.

Po vyvolání funkce *connect* čeká klient na odpověď od serveru, pokud dostane odpověď, bude do pole status vytisknuto hlášení „Connected", že odpověď nedostane, vytiskne chybové hlášení. Při úspěšném rozjetí komunikace musí klient čekat, dokud se nespustí LinuxCNC. Po úspěšném spuštění bude odesláno do klienta příkaz „Ready", čímž budou aktivována všechna tlačítka a bude připravena aplikace k provozu.

Data do serveru jsou posílána ve formátu struktury, skládající se ze tří čísel. Dvě čísla jsou typu double a představují souřadnice X a Y. Třetí číslo je typu int a určuje směr pohybu a děj pro osu Z (odběr podložky z pravé strany, ukládaní do levé atd). Data pro odeslání jsou připravována pomocí knihovny *struct* a metody *pack*. Celkový kód pro odeslání:

```
  command = (coords[0], coords[1], act)
  pack = struct.Struct('f f I')
  data = pack.pack(*command)
  sock.sendall(data)
```

První příkaz vytvoří balík pro odeslání, kde jsou uvedeny souřadnice a příkaz pro osu Z. Dále jsou specifikovány datové typy každého členu (*float, float, int*). Dalším příkazem na základě zvolených datových typů bude balíček převeden do bytové podoby. Posledním příkazem se vytvoří bytový příkaz, který bude odeslán do serveru.

7.3.3 Ukládaní nastaveni

Všechna provedená nastavení pomocí aplikace jsou ukládána do *.ini* souboru a budou automaticky načtena při každém spuštění programu. Práce s nastaveními je realizována pomocí knihovny *configparser*. Pro ukládání nastavení je použita funkce s následujícím kódem:

```
settings = configparser.ConfigParser()
settings.read(path)
settings.set(section, setting, value)
```

Pomocí dané funkce bude soubor s nastaveními otevřen a do určené sekce bude zapsán název a hodnota nastavení. Příklad uložených nastavení je vidět na obrázku 7.3.

```
1  [SETTINGS]
2  ip = 192.168.31.66
3  port = 7000
4  printers_num = 4
5  storages_num = 3
6
7  [PRINTERS]
8  printer_1 = X35.0 Y30.0 Front
9  printer_2 = X121.5 Y66.2 Back
10 printer_3 = X0.0 Y0.0 Front
11 printer_4 = X0.0 Y0.0 Front
12
13 [STORAGES]
14 storage_1 = X1.0 Y10.5 Front
15 storage_2 = X13.5 Y10.5 Back
16 storage_3 = X0.0 Y0.0 Front
17
```

Obrázek 7.6 Nastaveni pro klient

8 TESTOVANI POMOCI VIRTUALNIHO MODELU

8.1 Software pro PLC

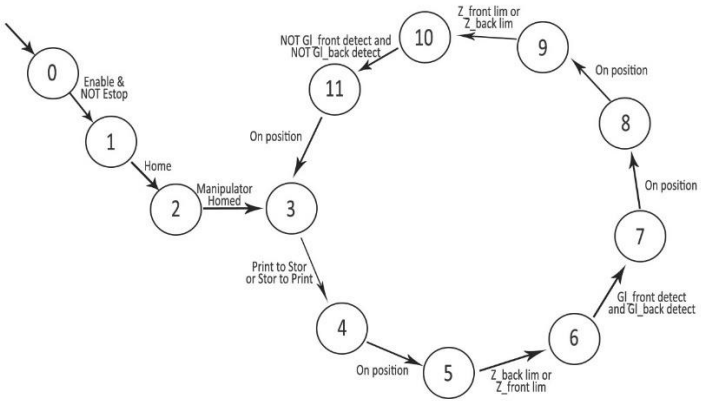
Virtuální model bude možné ovládat v automatickém a manuálním režimu. Prvním krokem byla vytvořena vizualizace pro ovládání manipulátoru (viz obrázek 8.1). Vizualizace má dvě obrazovky: první pro automatický režim, druhá pro manuální. Přepínání mezi obrazovkami je možné pomocí tlačítka Auto/Manual mode.

Obrázek 8.1 Vizualizace

Obě obrazovky obsahují tlačítka pro spuštění a houmování stroje a tlačítko Estop. V automatickém režimu je nutné vybrat číslo tiskárny a číslo skladu a poté jenom zvolit, bude-li tisková podložka umístěna ze skladu do tiskárny, nebo naopak z tiskárny do skladu. V manuálním režimu lze pomocí tlačítek pohybovat každou z os a ovládat pohyby dopravníku.

Program pro PLC byl napsán v jazyce SCL. Algoritmus práce je nastaven tak, že pro spuštění nemusí být stisknuto tlačítko Estop a musí být stisknuto tlačítko Enable, jinak nebude manipulátor pracovat. Jestli jsou obě podmínky splněny, bude aktivována možnost pohybu manipulátoru do nulové polohy. Pokud bude na základě snímačů určena nulová poloha, rozsvítí se indikátor Homed na obrazovce a bude umožněna práce s manipulátorem jak v automatickém, tak i v manuálním režimu.

Na obrázku 8.2 je ukázán stavový automat pro automatický režim manipulátoru. V tabulce 3 je uveden detailnější popis každého stavu. Program je realizován pomocí příkazu CASE, kde je na základě snímačů realizováno přepínání mezi stavy.



Obrázek 8.2 Stavovy automat

Tabulka 3 Detailnější popis stavu

Stav	Popis
0	Počáteční stav, manipulátor je neaktivní.
1	Manipulátor je zapnuty, čeká na příkaz pro návrat do nulové polohy.
2	Motory jsou zapnuty, manipulátor pohybuje směrem k nulové poloze, pokud nebude detekována pomoci snímačů.
3	Manipulátor je připraven k provozu, čeká na příkaz.
4	Manipulátor pohybuje na souřadnice zvolené tiskárny/skladu (místo odběru).
5	Vozík osy Z pohybuje dovnitř tiskárny/skladu, čeká na detekci krajní polohy vozíku.
6	Je zapnuty dopravník, čeká na detekci podložky.
7	Vozík osy Z vrací se na nulovou polohu.
8	Manipulátor pohybuje na souřadnice zvolené tiskárny/skladu (místo pro ukládáni podložky).
9	Vozík osy Z pohybuje dovnitř tiskárny/skladu, čeká na detekci krajní polohy vozíku.
10	Je zapnuty dopravník, čeká, pokud nebude odstraněna podložka.
11	Vozík osy Z vrací se na nulovou polohu.

8.2 Testováni vlastnosti

Hlavním úkolem simulace je vyzkoušení chování stroje při různých rychlostech a přibližně určit uspořádání budoucího skladu. Požadavkem na maximální rychlost je 0.4 m/s. Maximální zrychlení je 0.2 m/s2. Pro možnost zadání těchto parametrů do MCD je nutné přepočítat rychlost na °/s a zrychlení na °/s2. Víme, že za jednu otáčku motoru osa X projezdí 37.7 mm a osa Y 18.85 mm. Proto:

$$v_{maxX^\circ} = \left(\frac{v}{37.7mm}\right) * 360^\circ = \left(\frac{400}{37.7}\right) * 360^\circ = 3819.6\ ^\circ/s \quad (15)$$

$$v_{maxY^\circ} = \left(\frac{v}{18.85mm}\right) * 360^\circ = \left(\frac{400}{18.85}\right) * 360^\circ = 7639.3\ ^\circ/s \quad (16)$$

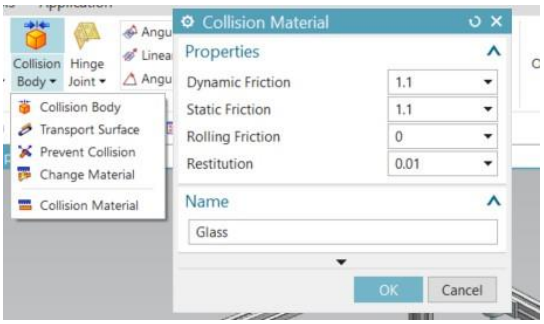
Stejným způsobem přepočítáme zrychleni:

$$a_{maxX^\circ} = \left(\frac{a}{37.7mm}\right) * 360^\circ = \left(\frac{200}{37.7}\right) * 360^\circ = 1909\ ^\circ/s^2 \quad (17)$$

$$a_{maxY^\circ} = \left(\frac{a}{18.85mm}\right) * 360^\circ = \left(\frac{200}{18.85}\right) * 360^\circ = 3819\ ^\circ/s^2 \quad (18)$$

Vypočtené parametry je nutné zadat do funkce *Position Control* (obrázek 6.9).

Dalším krokem pro větší přesnost simulace by bylo vhodné nastavit správné součinitele tření pro tiskovou podložku (sklo) a pásový dopravník (guma). Pro nastavení daných parametrů je potřeba vytvořit nový kolizní materiál. Toho lze docílit pomocí funkce *Collision Material*.

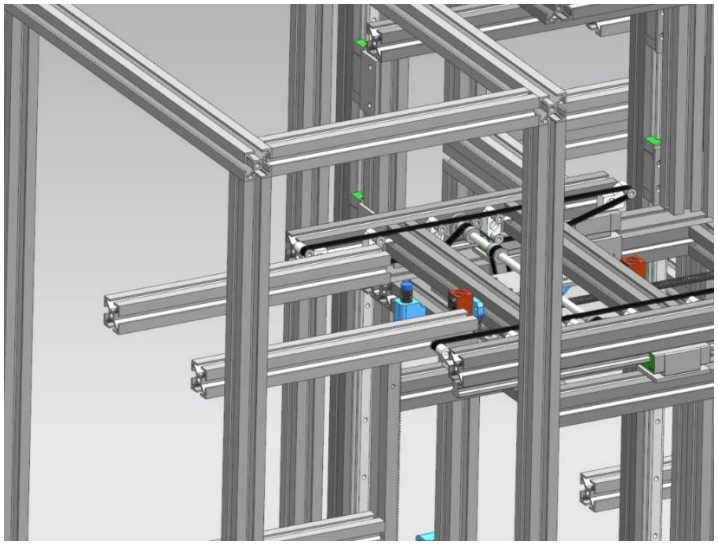


Obrázek 8.3 Nastaveni kolizního materiálu

Následně byl zjištěn součinitel tření mezi sklem a gumou, který v ideálních podmínkách dosahuje až 1.2. S ohledem na to, že manipulátor nebude pracovat v ideálních podmínkách, byl zvolen součinitel 1.1. Po nastavení daných parametrů je možné přejít k testování [21].

Simulace byla na začátku provedena pro zrychlení 0.2 m/s2 a s prázdnou tiskovou podložkou. Během pohybu žádný problém nevznikal, podložka se neposouvala na vozíku a lze říct, že pro prázdnou podložku můžeme takovou rychlost a zrychlení používat. Dalším krokem byla simulace provedena pro neprázdnou podložku. Pro imitace vytištěného dílu na podložce byla zvětšena hmotnost podložky o 0.5 kg, což by měla být maximální hmotnost tisku. Po daných úpravách bylo provedeno několik pokusů přemisťování podložky. Během této simulace také nevznikl žádný problém, z čehož lze vytvořit výsledek se správně zvoleným zrychlením a rychlostí pohybu.

Dále bylo provedeno několik dalších simulací s postupným zvětšením zrychlení, jejichž cílem bylo nalézt mezní zrychlení, kdy se začne podložka na vozíku posouvat. Proto pro každou další simulaci bylo zrychlení zvětšeno o 0.05 m/s2 a provedeno několik pokusů přemisťování pro prázdnou podložku a podložku s tiskem. Bylo zjištěno, že zrychlení lze používat až po 0.35 m/s2. Při zrychlení 0.35 m/s2 se podložka u zastavení pohybu začala posouvat.



Obrázek 8.4 Uspořádaní skladu pro tiskové podložky

Další otázkou bylo uspořádání skladu. Vzorový model, který byl vytvořen na začátku, umožňoval ukládání podložky, ale ne její odběr. Z toho hlediska, že sklad by neměl mít dopravníky, aby nebyl příliš drahý, je nutné rozmyslet takovou konstrukci, která by umožnila odběr podložek jen s využitím manipulátoru. Po vyzkoušení několika možností bylo rozhodnuto, že nejlepší variantou je realizovat sklad tak, aby byly podložky ukládané co nejblíž k čelní straně skladu a aby mohl vozík podjet pod podložku, jak to je ukázáno na obrázku 8.4. Takovým způsobem jde realizovat odběr bez využití pomocných dopravníků.

ZÁVĚR

Cílem práce bylo navrhnout konstrukce stroje pro obsluhování 3D tiskáren. Prvním krokem byl definován princip funkčnosti stroje z hlediska mechaniky, byly provedeny výpočty parametrů komponentů a následně byly vybrány potřebné komponenty. Stroj Bylo rozhodnuto stroj realizovat jako 3-osý CNC manipulátor, umístěný na čelní straně tiskáren. Pro jeho stavbu jsou využity hliníkové stavebnicové profily o rozměru 30 × 30 mm. Lineární pohyb je zajištěn pomocí prizmatického vedení Hiwin HGR15 a páru ozubených hřebenů s ozubeným kolem. Pro realizace pohybu jsou využity motory Nema23 se zpětnou vazbou.

S využitím vybraných součástek v programu Siemens NX byl vytvořen 3D model konstrukce manipulátoru, podle kterého bude sestaven prototyp stroje. Vytvořený 3D model byl také využit pro simulace provozu manipulátoru v programu Siemens Mechatronix Concept Designer. Pomocí simulace byla otestována funkčnost stroje, otestovány byly maximální rychlosti a zrychlení pohybu a bylo určeno uspořádání skladu pro tiskové podložky. Řízení virtuálního pohyblivého modelu bylo realizováno pomocí virtuálního PLC vytvořeného v programu PLCSIM Advance.

Pomocí programu WSCAD byl navržen elektrický rozvod stroje. Pro jednoduchost realizace komunikace a připojení snímačů byly navrženy desky plošných spojů pro Arduino a Beaglebone.

Na Beaglebone byl nainstalován operační systém Macinekit a pro něj jsou vytvořeny konfigurační soubory a Python script pro ovládání. Dalším krokem byl vytvořen software pro počítač (klientská aplikace), který umožní ovládání manipulátoru. Arduino byla naprogramovaná na řízení mechanismu odběru podložky. Součástí bylo vytvoření komunikace mezi těmito komponentami.

Vyvinutý stroj umožňuje automatizovat odběr vytištěných dílů a spuštění dalšího tisku 3D tiskáren a tím ušetřit čas a náklady na jejich obsluhu. Jelikož jsou komponenty voleny z hobby a semi-profi sféry, nelze ještě s jistotou vyčíslit míru finančního přínosu. Nesporně lze ale vyzdvihnout výhodu možnosti rozšíření pracovního pole pro větší počet tiskáren.

Literatura

[1] *Šablona pro BP/DP a prezentace v2.63* [online]. Brno: FEKT VUT, 2017 [cit. 2017-03-06]. Dostupné z: http://latex.feec.vutbr.cz/sablona/

[2] *Hiwin HGR15* [online]. [cit. 2019-12-15]. Dostupné z: https://www.hiwin.com/linear-guideways.html

[3] *Linear Guideway SBR-16* [online]. [cit. 2019-12-15]. Dostupné z: https://www.alibaba.com/product-detail/abba-linear-guide-SBR16-SBR-16_60221325042.html

[4] *Ozubené hřebeny a kola* [online]. [cit. 2019-12-15]. Dostupné z: http://www.cncshop.cz/ozubene-hrebeny-kola_c

[5] *Řemenová převodovka* [online]. [cit. 2020-04-22]. Dostupné z: https://www.amazon.es/SUMRAY-HTD5M-reducci%C3%B3n-cintur%C3%B3n-dentado/dp/B07FD8NH35

[6] *Hliníkový profil* [online]. [cit. 2020-04-22]. Dostupné z: https://www.kanya.com/en/products/extrusion-connecting-system/33-18042019-075759/b02-1-02-02-200.html

[7] *Nema 23 Closed Loop Stepper Motor* [online]. [cit. 2019-12-22]. Dostupné z: https://www.omc-stepperonline.com/nema-23-closed-loop-stepper-motor-185nm-26203ozin-encoder-1000cpr

[8] *Makerbase Nema 23 Stepper Motor* [online]. [cit. 2019-12-22]. Dostupné z: http://www.chinaexpress.cz/cs/item/3d-printer-parts-closed-loop-nema23-stepper-motor-mks-servo57a-nema-23-servo-motor-prevents-losing-steps/33004356135.html

[9] *Beaglebone Black* [online]. [cit. 2019-12-22]. Dostupné z: https://www.microcenter.com/product/469274/beagleboard-beaglebone-black-rev-c-board

[10] *DM506-EC Stepper Motor Driver* [online]. [cit. 2019-12-22]. Dostupné z: https://www.amazon.com/Stepper-EtherCAT-Protocol-DC24-50V-1-5-5-6A/dp/B07XLJN69J

[11] *TB6600 Stepper Motor Driver* [online]. [cit. 2019-12-22]. Dostupné z: https://www.les3dprinter.com/products/hanpose-tb6600-stepper-driver-for-nema-17-23-34-stepper-motor-cnc-engraving-machine

[12] *Napájecí zdroje PS100-500* [online]. [cit. 2019-12-22]. Dostupné z: http://www.cncshop.cz/napajeci-zdroje_c

[13] *Průmyslové spínané napájecí zdroje* [online]. [cit. 2019-12-22]. Dostupné z: http://www.cncshop.cz/spinane-zdroje_c

[14] *Hiwin Linear Guideways* [datasheet]. [cit. 2019-12-22]. Dostupné z: https://www.hiwin.com/pdf/linear_guideways.pdf

[15] *Arduino Nano* [online]. [cit. 2019-12-22]. Dostupné z: https://store.arduino.cc/arduino-nano

[16] *Nema23 datasheet* [datasheet]. [cit. 2019-12-22]. Dostupné z: https://www.omc-stepperonline.com/download/23HS20-2004S.pdf
[17] *Beaglebone Black datasheet* [datasheet]. [cit. 2019-12-24]. Dostupné z: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf
[18] *Pohony s krokovými motorky* [online]. [cit. 2019-12-24]. Dostupné z: http://fei1.vsb.cz/kat410/studium/studijni_materialy/se/cast_C_el_pohony/se_eph_c1_krokac_02_teorie.pdf
[19] *Mikrospínač MSW-0* [online]. [cit. 2019-12-24]. Dostupné z: http://www.cncshop.cz/mikrospinac-msw-0
[20] *Indukční snímače* [online]. [cit. 2019-12-24]. Dostupné z: http://www.cncshop.cz/indukcni-snimace_c
[21] *Adhesion and friction between glass and rubber* [online]. [cit. 2020-04-22]. Dostupné z: https://pubs.rsc.org/en/content/articlelanding/2018/sm/c8sm00847g#!divAbstract

Seznam symbolů, veličin a zkratek

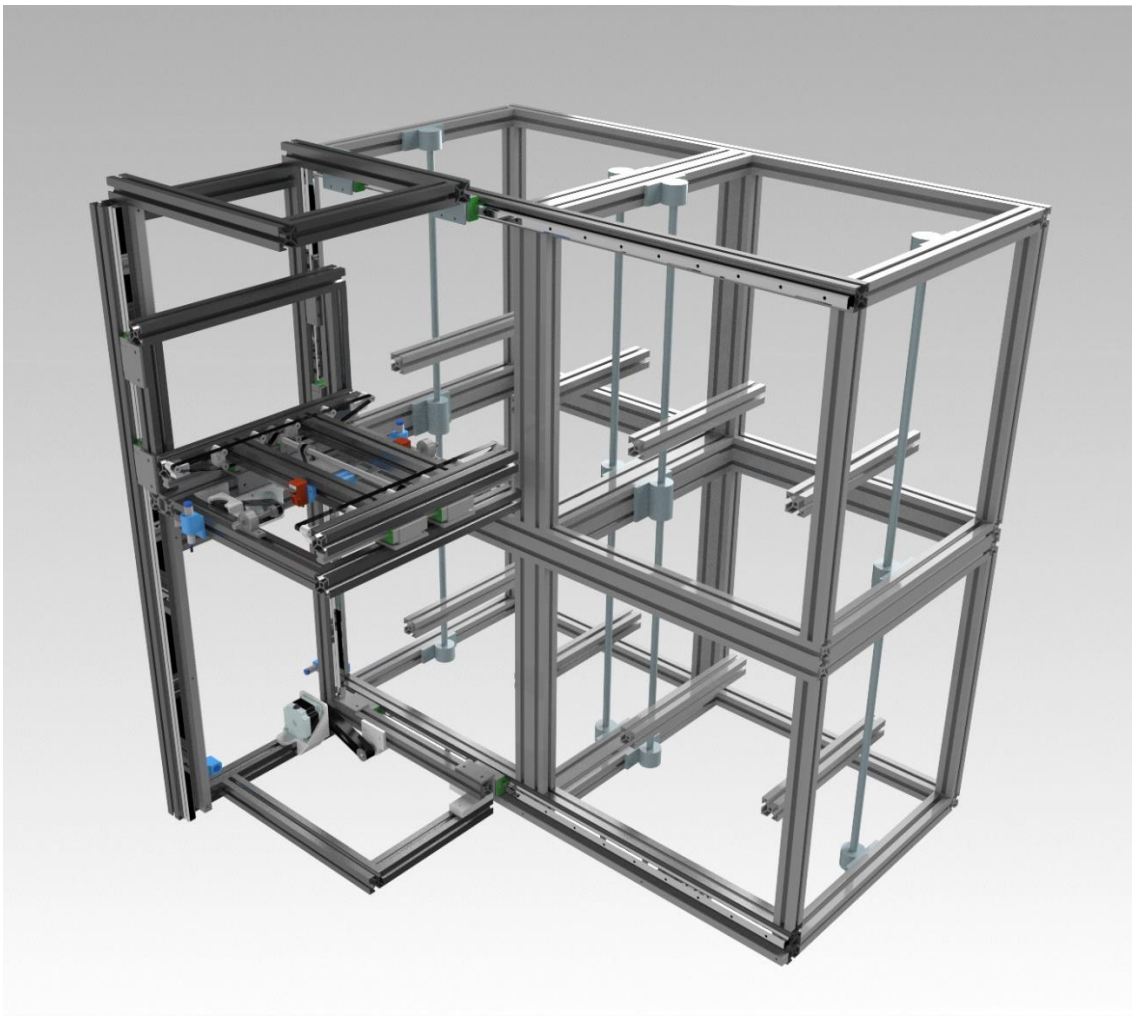
FEKT	-	Fakulta elektrotechniky a komunikačních technologií
VUT	-	Vysoké učení technické v Brně
CNC	-	Computer numerical control
PC	-	Personal computer
PWM	-	Pulse width modulation
MCD	-	Mechatronics Concept Designer
PLC	-	Programmable Logic Controller

v	-	rychlost	[m/s]
a	-	zrychleni	[m/s^2]
g	-	tíhové zrychleni	[m/s^2]
m	-	hmotnost	[kg]
F	-	sila	[N]
M	-	moment sily	[N.m]
d	-	průměr	[m]
I	-	proud	[A]

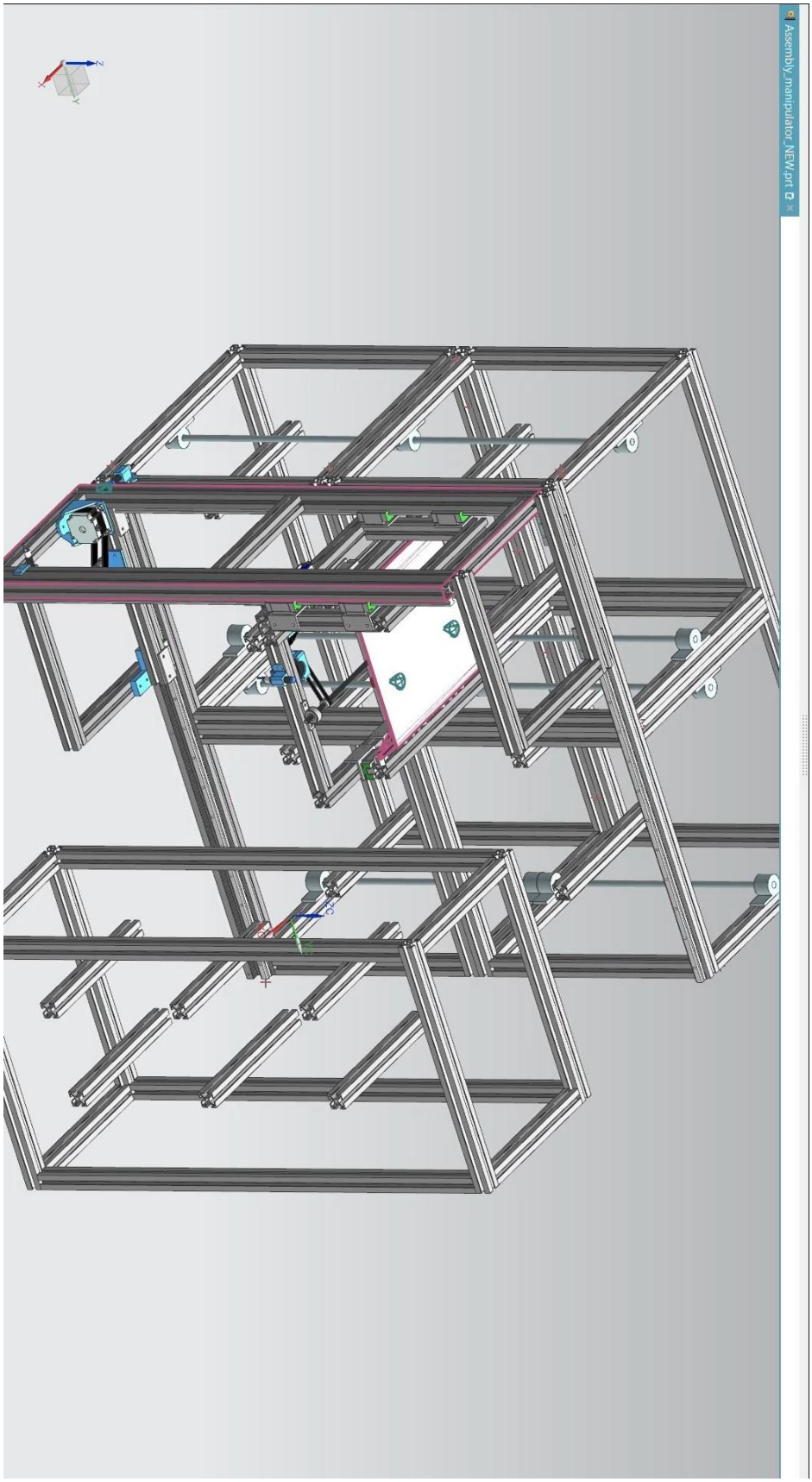
Seznam příloh

Přílohy

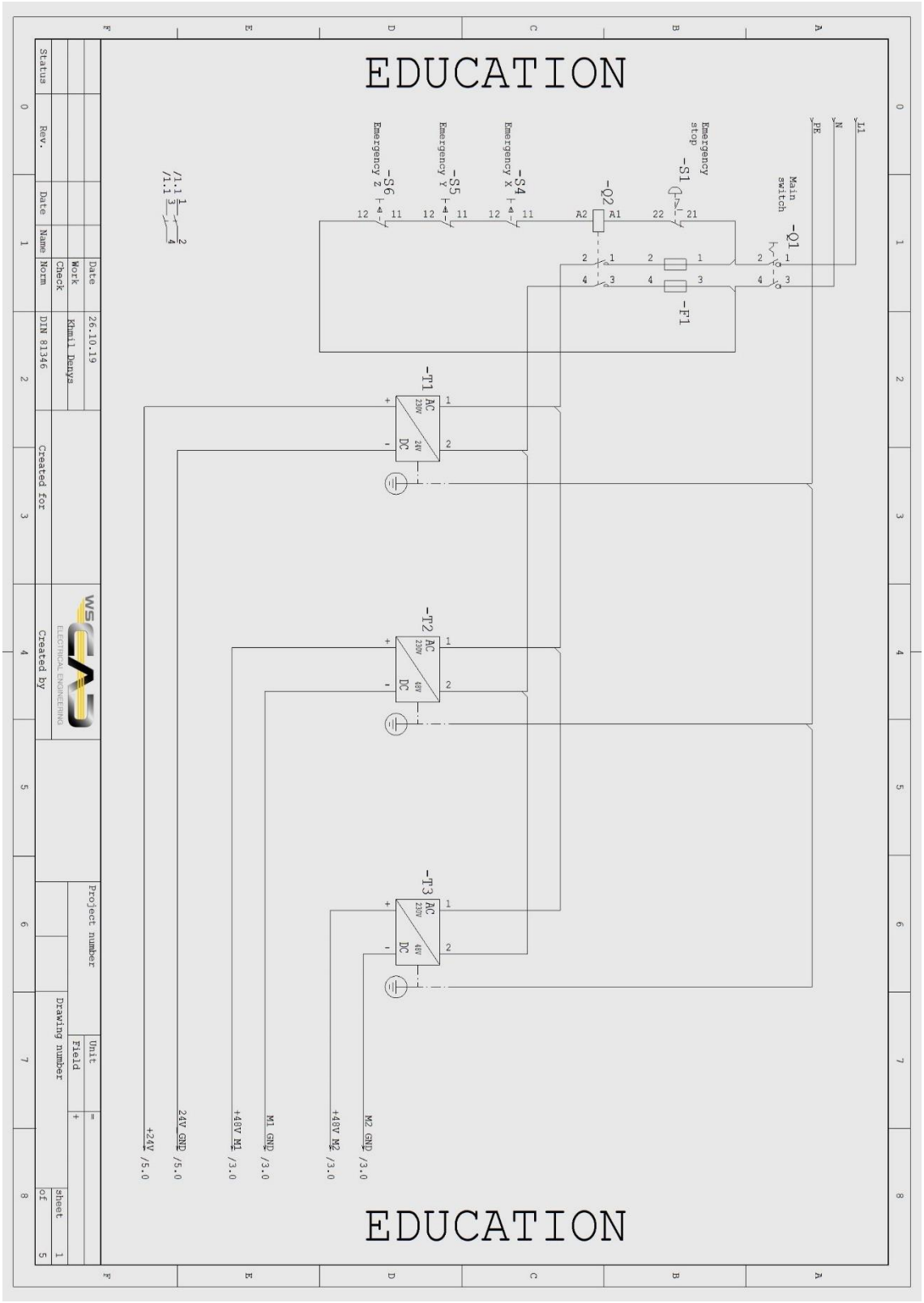
Příloha 1 3D model manipulátoru



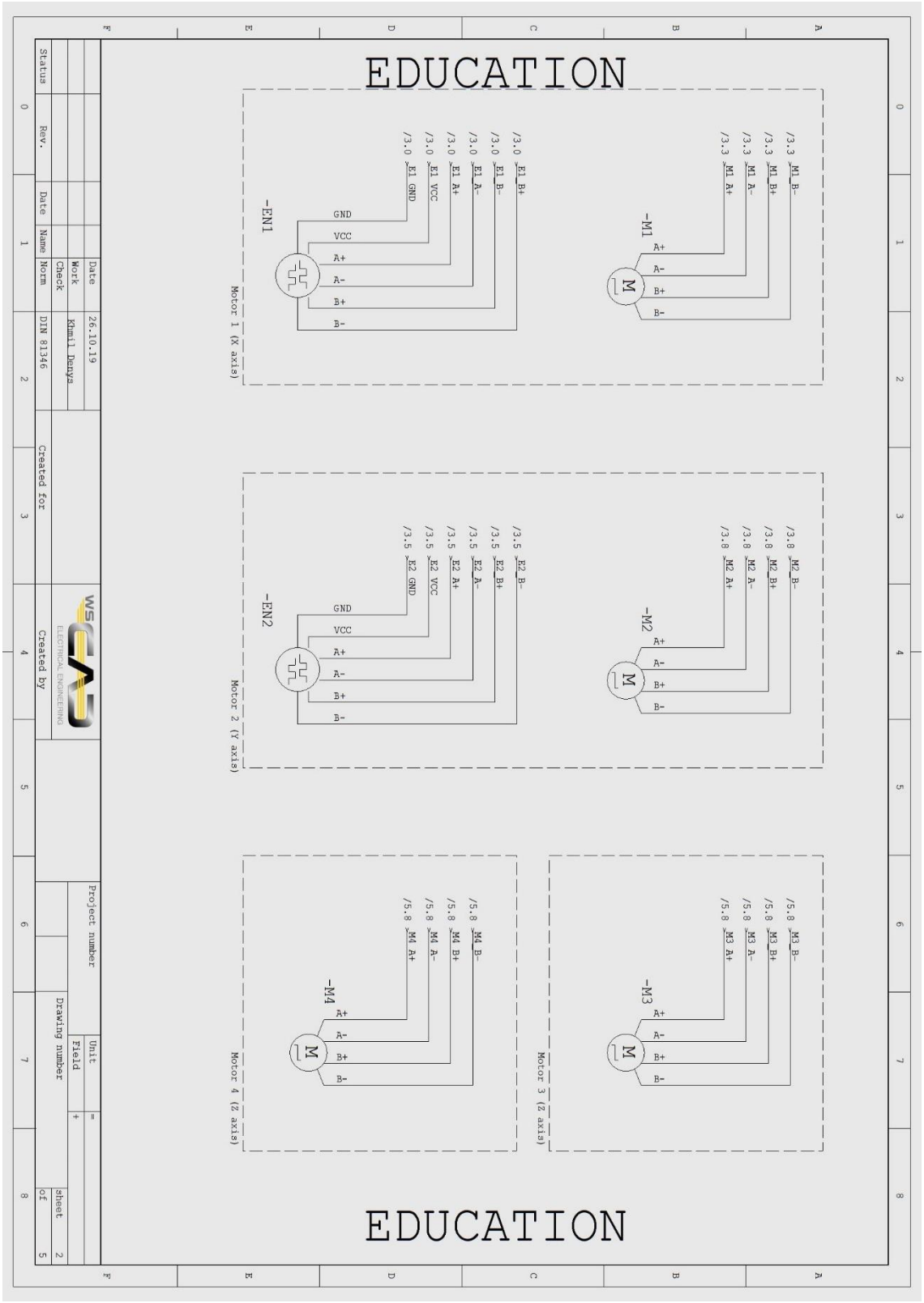
Příloha 2 Virtuální pohyblivý model



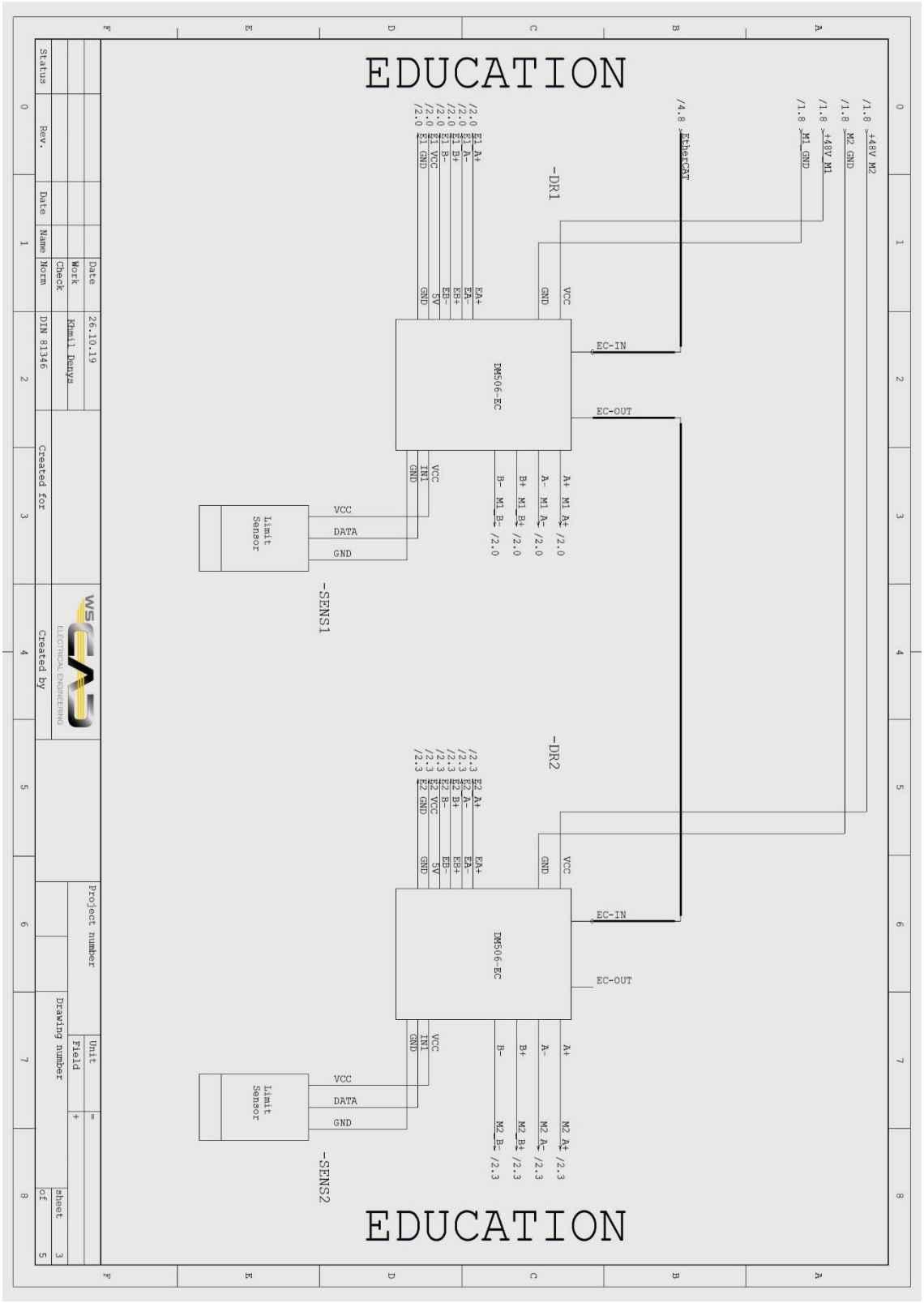
Příloha 3 Schéma elektrického rozvodu str. 1



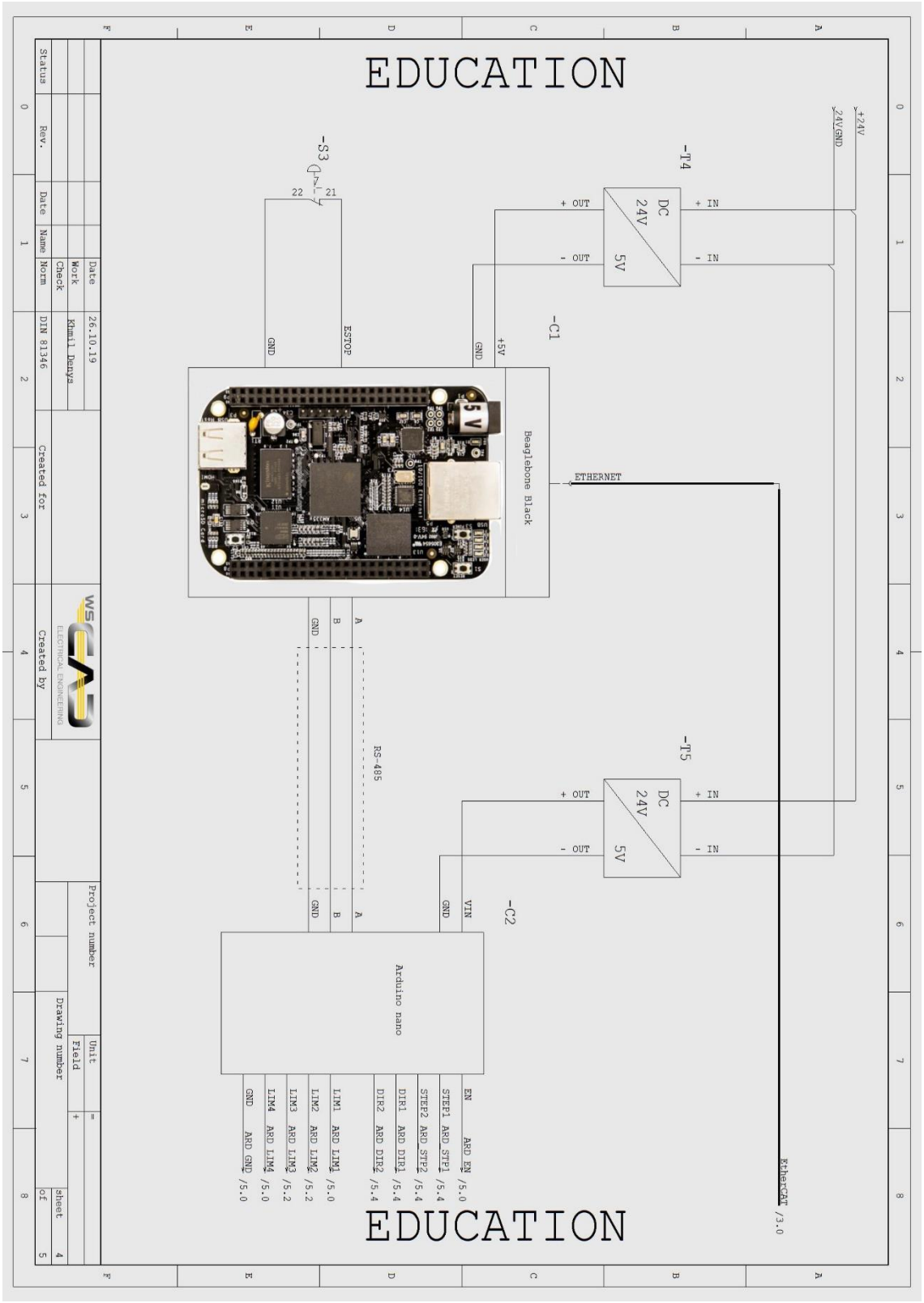
Příloha 4 Schéma elektrického rozvodu str. 2



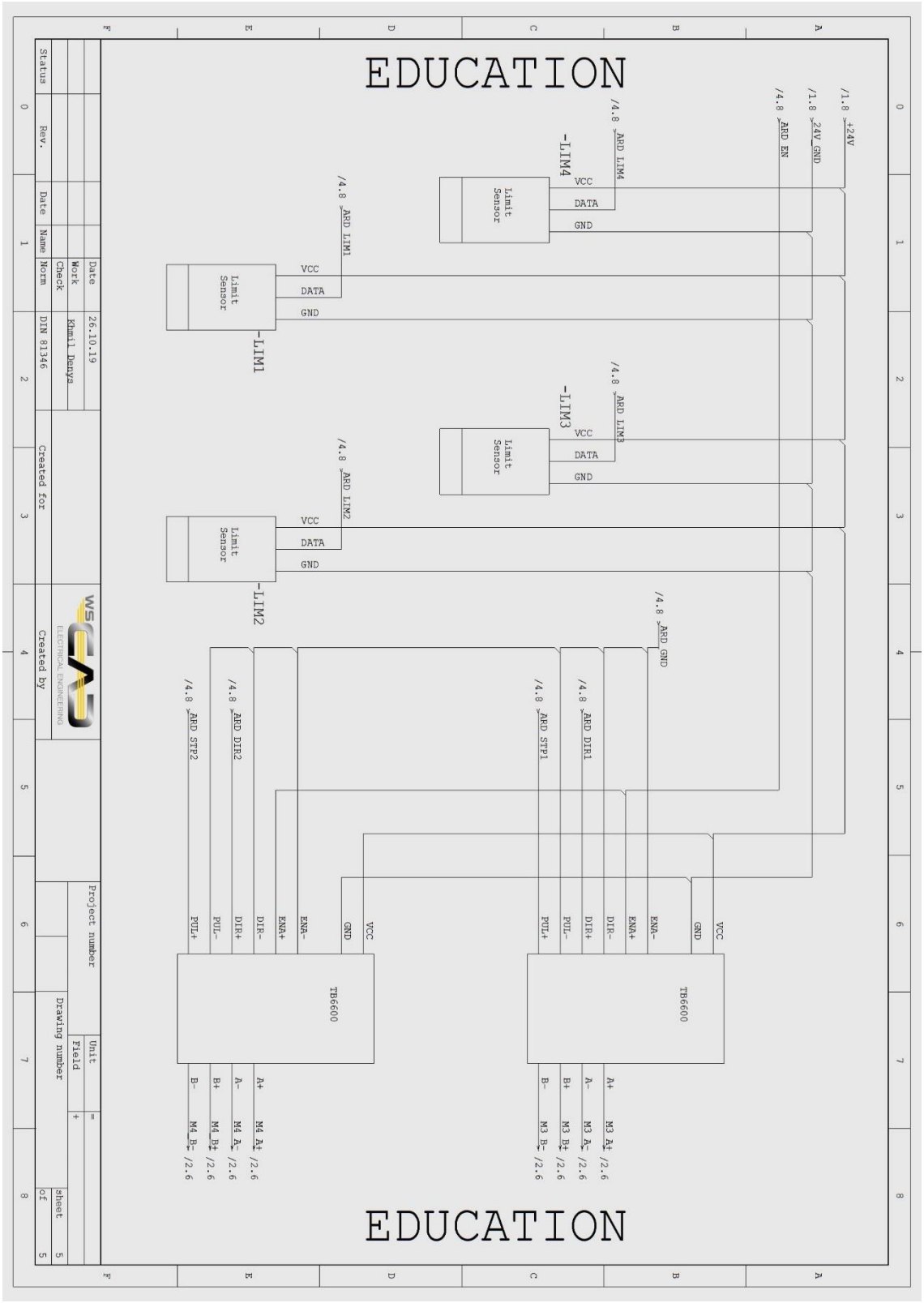
Příloha 5 Schéma elektrického rozvodu str. 3



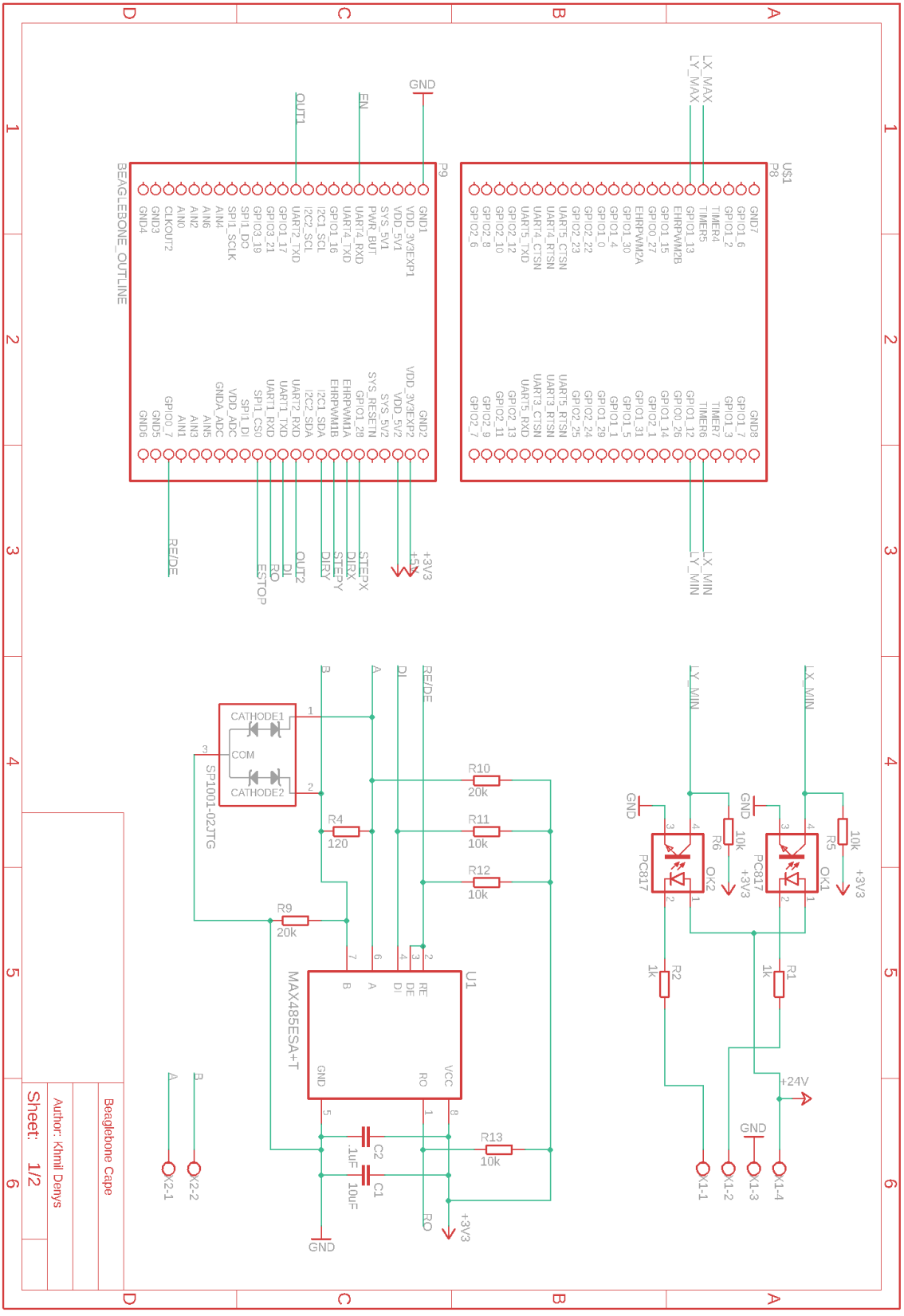
Příloha 6 Schéma elektrického rozvodu str. 4



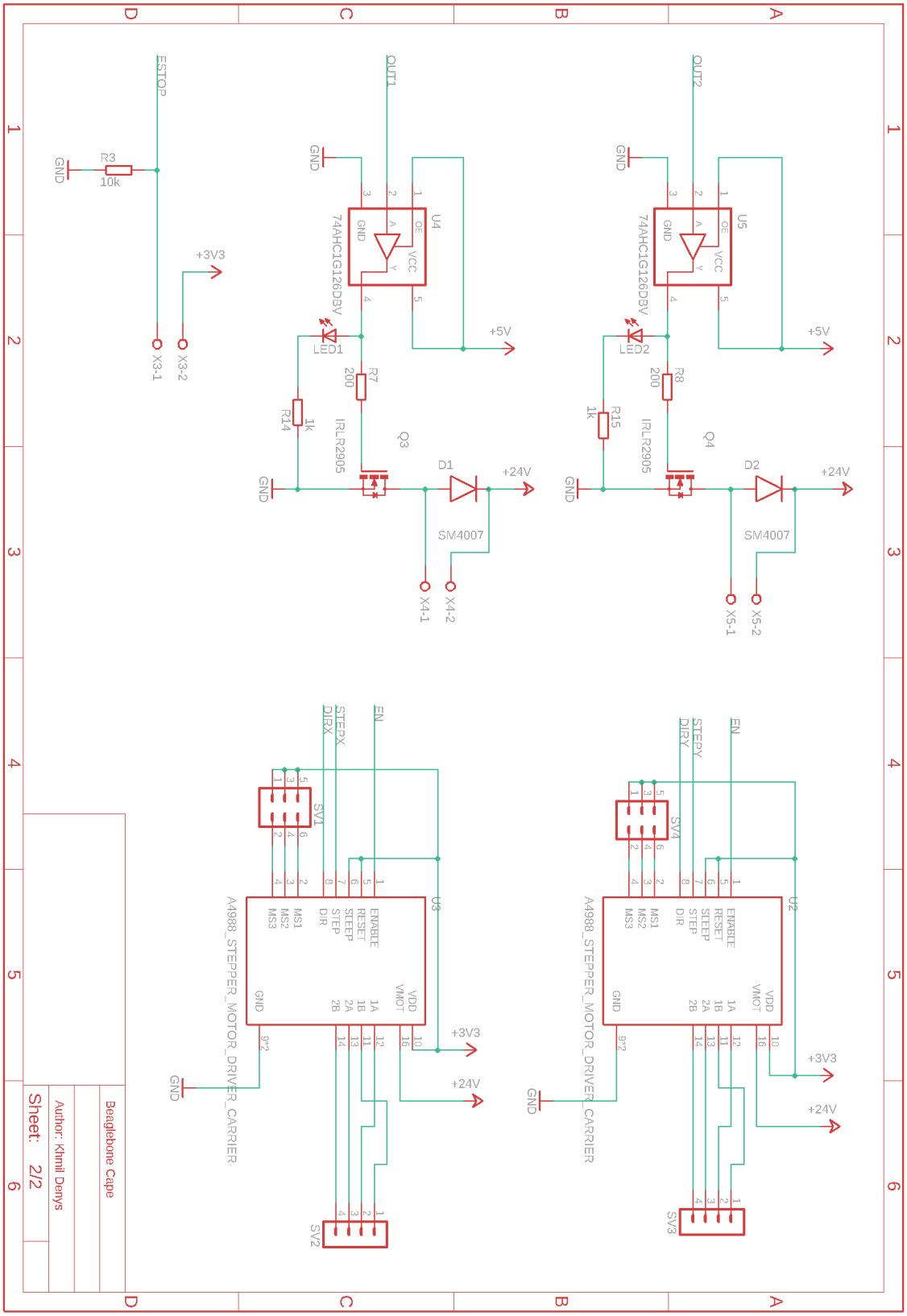
Příloha 7 Schéma elektrického rozvodu str. 5



Příloha 8 Schéma DPS pro Beaglebone str. 1



Příloha 9 Schéma DPS pro Beaglebone str. 2



Příloha 10 Schéma DPS pro Arduino